Timo Poranen (ed.)

Software Projects 2010-2011

UNIVERSITY OF TAMPERE
SCHOOL OF INFORMATION SCIENCES
REPORTS IN INFORMATION SCIENCES 2

TAMPERE 2011

UNIVERSITY OF TAMPERE
SCHOOL OF INFORMATION SCIENCES
REPORTS IN INFORMATION SCIENCES 2
AUGUST 2011

Timo Poranen (ed.)

Software Projects 2010-2011

SCHOOL OF INFORMATION SCIENCES
FIN-33014 UNIVERSITY OF TAMPERE

ISBN 978-951-44-8510-7

ISSN-L 1799-8158
ISSN 1799-8158

Preface

This report contains project stories of 13 software development projects and one usability team. The students came from Project Work and Software Project Management courses. The stories describe what kind of experiences groups got during their project and what was the software product that came out from the project. In the end of each story there are project statistics.

Table 1 gives an overview of projects, project type (WWW, mobile or standalone application), client (Dept.= CS department, Assoc.= non-commercial associations, Company, Demola or Other non-commercial client), development model, group size (number of managers + number of developers + number of usability team members) and working hours of the project. Project types which are marked with an asterisk (*), developed further an existing software.

Table 1: General project statistics.

Project	Type	Client	Dev. Mod.	Group	Hours
Uteam	-	CS Dept.	-	2+11	2167
Tapahtumakalenteri	WWW	Assoc.	Scrum	2+3+1	1090
Majava3	WWW*	CS Dept.	Scrum	2+5+1	1205
TOC	WWW	Company	Scrum	2+5+1	1168
Vaalilupausarkisto	WWW	Assoc.	Scrum	2+5+1	1759
Luuppi	WWW	Assoc.	Scrum	2+4+1	1118
MAOP	WWW	Other	Scrum	2+5+1	1195
MoTiPe	Appl.	CS Dept.	Scrum	2+5+1	1104
OhOp	WWW	CS Dept.	Scrum	2+3+1	1298
InputDevice	Appl.	CS Dept.	Iterative	2+4+1	1022
Commander	Appl.	Demola	Scrum	2+6+1	1600
Ubiquity	WWW+mobile	Demola	Scrum	2+5+1	1345
MobileTransport	Mobile	Demola	Scrum	1+3+1	1063
TamBiC2	WWW*	Univ.	Iterative	2+5+1	1023

Although 11 projects applied Scrum development model, they had one major difference when compared to standard Scrum: daily scrum meetings were mainly organized virtually using IRC or similar real-time messaging systems, or the daily meetings were omitted.

Table 2 contains general course statistics (number of projects and usability teams, number of students in the courses and average project size in working hours) starting from year 2005.

Table 2: Course statistics 2005-2011

Academic year	Projects	Usability teams	PW students	SPM students	Average project size
2005-6	19	1	98	8	1008
2006-7	18	2	87	34	1089
2007-8	14	1	70	29	997
2008-9	10	1	60	39	1643
2009-10	15	1	80	34	1151
2010-11	13	1	70	27	1230

During the course we used projectWiki for maintaining course and project related documentation: https://projectwiki.cs.uta.fi. The wiki also contains some articles on project management and project management tools, including lists of end-products currently in use, course related publications and course related videos:

- https://projectwiki.cs.uta.fi/wiki/Finished_projects
- https://projectwiki.cs.uta.fi/wiki/Course_publications
- https://projectwiki.cs.uta.fi/wiki/List_of_project_videos

Course staff thanks our clients and students for great projects!

Timo Poranen

Tampere, August 2011

UTeam

Yleiskuvaus

UTeamin eli käytettävyysryhmän ajatuksena oli tarjota muille projektityöryhmille käytettävyysosaamista vuorovaikutteisen teknologian opiskelijoiden muodossa. Lisäksi käytettävyysosaamisen keskittäminen yhden projektin alle tarjosi asiakasprojekteissa työskenteleville käytettävyysasiantuntijoille vertaistukea.

Jokainen UTeamin jäsen toimi yhdessä tai useammassa asiakasprojektissa käytettävyyseksperttinä. UTeamin jäsenet toimivat heille määrätyissä asiakasprojekteissa itsenäisesti. Projektinjohdon rooli oli organisoida ja tukea tätä toimintaa.

Organisointi ja hallinta

UTeamin asiakkaina toimivat projektityökurssin 2010-2011 ohjelmistoprojektit.

UTeam koostui kahdesta projektipäälliköstä ja yhdestätoista jäsenestä.

Projektipäälliköinä toimivat:

Kristiina Niskala

Piia Sajasalo

Jäsenet olivat:

Jenni Paukkunen (Vaalilupausarkisto, Majava3)

Antti Suoninen (MobileTransport)

Jussi Palomäki (Ubiquitous, Unreal Input Testing)

Lasse Mäkinen (Silverlight)

Markus Ijäs (Joomla Tapahtumakalenteri)

Miika Mikkola (HP+)

Mikko Pajulahti (TamBic2)

Minna Hara (MoTiPe)

Niina Hakala (Ohop)

Saana Riihelä (Maop)

Saila Oldén (Luuppi)

Kuva 1: Uteam vuonna 2010-11. Takana (vas.) Saana, Saila, Mikko, Lasse, Antti ja Jenni. Eturivissä projektipäälliköt Kristiina ja Piia. Poissa Jussi, Markus, Miika, Minna ja Niina.

Projektipäälliköt hoitivat työt yhteistuumin, eikä tehtäviä jaettu varsinaisesti erillisiin vastuualuisiin. Kristiina keskittyi kuitenkin enemmän henkilöstöhallinnollisiin tehtäviin ja Piia teknisen projektipäällikön tehtäviin.

Menetelmät ja työkalut

Projektin luonteen takia tiimillä oli käytössään paljon erilaisia työkaluja. UTeamin sisäisessä toiminnassa käytettiin Doodle-kalenteriohjelmaa, sähköpostia, tiimin omia verkkosivuja sekä Google Documents-palvelua. Tiimin projektipäälliköt kommunikoivat paljon myös Skypen avulla. Ohessa listattuna muita UTeamiläisten kurssilla käyttämiä ohjelmistoja ja työkaluja.

Kuvankäsittely:

- Photoshop

Tekstinkäsittely:

- Microsoft Word
- WordPad
- Notepad
- OpenOffice Writer

Versionhallinta:

- Mercurial

Kommunikointi:

- IRC
- Jabber
- Tekstiviesti
- Wiki

Koodaus:

- Microsoft Visual Studio 2010
- Microsoft Expression Blend 4
- Qt Creator IDE
- Qt Editor

Muut:

- Wiki
- Redmine
- Microsoft Excel
- Käytettävyyslaboratorio
- Kyniä ja paperia
- FileZilla

Projektin vaiheet ja kehitysmalli

Projektin luonteesta johtuen käytössä ei ollut mitään varsinaista kehitysmallia. UTeamilla ei ollut käytössä säännöllisiä tapaamisia, koska pyrimme pitämään oman organisaatiomme mahdollisimman kevyenä. Projektipäälliköt organisoivat tapaamisia ja jakoivat tehtäviä tiimiläisille tarpeen mukaan. Lisäksi tiimiläisten kuulumisia kyseltiin sähköpostitse säännöllisin väliajoin.

UTeamin tärkeitä päivämääriä:

15.9.2010	Projektin esittely muille projektipäälliköille
23.9.2010	Uteamin ensimmäinen tapaaminen
11.10.2010	Esitutkimuksen katselmointi
5.11.2010	Projektisuunnitelman katselmointi
1.12.2010	Projektiesitys
14.12.2010	Työpaja ryhmä 1
15.12.2010	Työpaja ryhmät 2 ja 3
16.12.2010	Käytettävyyslaboratorion esittely
27.1.2011	Pikkujoulut
25.2.2011	Loppuesitys
31.3.2011	Lopputapaaminen ja tiimin viimeinen palaveri

Kokemuksia

Alla on listattuna riskitekijöitä, joita tavalla tai toisella kohtasimme projektin aikana. Riskitekijät on lajiteltu projektisuunnitelmassa esiintyneiden kategorioiden mukaan.

- Teknologiset riskit
    - Toisen projektipäällikön henkilökohtainen kannettava tietokone tilttasi kesken projektin, eikä suostu enää aukeamaan. Onneksi kaikki koneella olleet projektia koskevat tiedostot oli tallennettu myös

muualle, joten vahinkoa ei päässyt syntymään -ainakaan projektin osalta.

- Henkilöstöriskit
  - Muiden projektien ryhmäläisten lopettaminen alensi myös jäljelle jääneiden ryhmäläisten motivaatiota. Onneksi kukaan UTeamilainen ei jättänyt projektia kesken!
  - Projektien pitkittyessä motivaatio laski.
  - Osassa projekteista myös UTeamilaisten työmäärät kasvoivat kun ryhmäläisiä lopetti projektityöskentelyn kesken.

- Organisaatioriskit
  - Osalla asiakkaiden edustajista tuntui oleva motivaation puutetta.
  - Erityisesti osassa monikansallisista projekteista ilmeni kommunikaatio-ongelmia ja informaatiokatkoksia.
  - Osa projekteista ei toimittanut UTeamin projektipäälliköille viikkoraporttejaan useista pyynnöistä huolimatta. Näin ollen tiimin vetäjät eivät voineet seurata projektin kulkua.
  - Suurta ryhmää oli hankala saada palaveriin yhtäaikaa. Pyrimme kuitenkin sopimaan palaverit niin, että palaverista ei puuttuisi aina samat ihmiset. Tässä onnistuttiin melko hyvin. Palaveriajan sopimisessa käytetty Doodle-työkalu toimi hyvin.

- Työkaluriskit
  - Laitteiden ja tarvittavien tilojen käyttöönoton viivästymiset aiheuttivat aikatauluongelmia projekteissa.

- Vaatimusmäärittelyriskit
  - Muutamassa projektissa käyttöliittymää ja ohjelmiston toiminnallisuutta jouduttiin karsimaan niin paljon, ettei ollut mielekästä tehdä varsinaisia käytettävyystestejä. Käytettävyyttä pyrittiin arvioimaan kevyemmillä menetelmillä. Ketterien menetelmien käyttö pienensi kuitenkin muutoksista aiheutuneita seurauksia.

- Arviointiriskit
  - Suurin osa tiimin asiakasprojekteista oli vähintään kaksi viikkoa alkuperäistä aikataulua myöhässä. Muutama projekti vielä enemmän. Tämä viivästytti käytettävyystiimin työskentelyn päättämistä.

- Eettiset ja moraaliset riskit
  - Näitä riskejä ei kohdattu projektin aikana.

Muut projektit suhtautuivat käytettävyysasiantuntijoihin pääsääntöisesti hyvin, mutta joitakin erimielisyyksiä ilmeni projektien edetessä. Joissakin projekteissa ei tunnuttu ymmärtävän käytettävyysasiantuntijan roolia ja käytettävyyden tärkeyttä.

Tilastotietoa

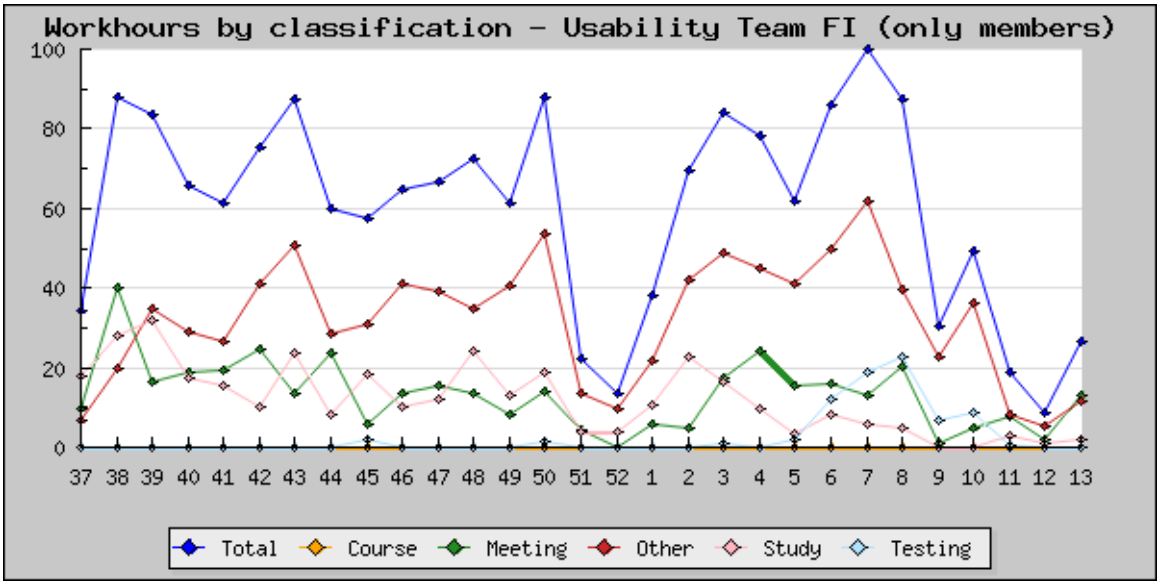
Alla olevissa taulukoissa on esitetty tiimin työtuntien jakautumista eri projektien kesken sekä projektiin käytettyjen viikkojen kesken.

Ryhmän koko	Kehitysmalli	Aloituspäivä	Lopetuspäivä	Päiviä yhteensä	Tunteja yhteensä
2+11	ei käytössä	10.9.2010	31.3.2011	210	2167

Taulukko 1: Yleistä tietoa projektista.

Projekti	Tunnit	%
Suunnittelu ja johtaminen	400	18,5
UTeam tiimiläiset	422,5	19,5
VLA	102	4,5
HP+	90	4,0
Majava 3	78,5	3,5
Ubiquitous	127,5	6,0
Tapahtuma-kalenteri	190,5	8,5
TamBic2	108	5,0
MoTiPe	107,5	5,0
Mobile Transport	141	6,5
Luuppi	70,5	3,0
UIT	26	1,0
OhOp	93	4,0
Silverlight	124	5,5
MaOp	86	4,0
UTeam yhteensä	2167	100

Taulukko 2: Käytettävyystiimin työtunnit per projekti.



Diagrammi 1: Käytettävyystiimin työtunnit per viikko.

Dokumentti	Sivuja	Versiot
Esitutkimus	10	1.0
Projektisuunnitelma	34	1.4
Loppuraportti	34	1.0
Project story	6	1.0
Viikkoraportit	27 kpl	

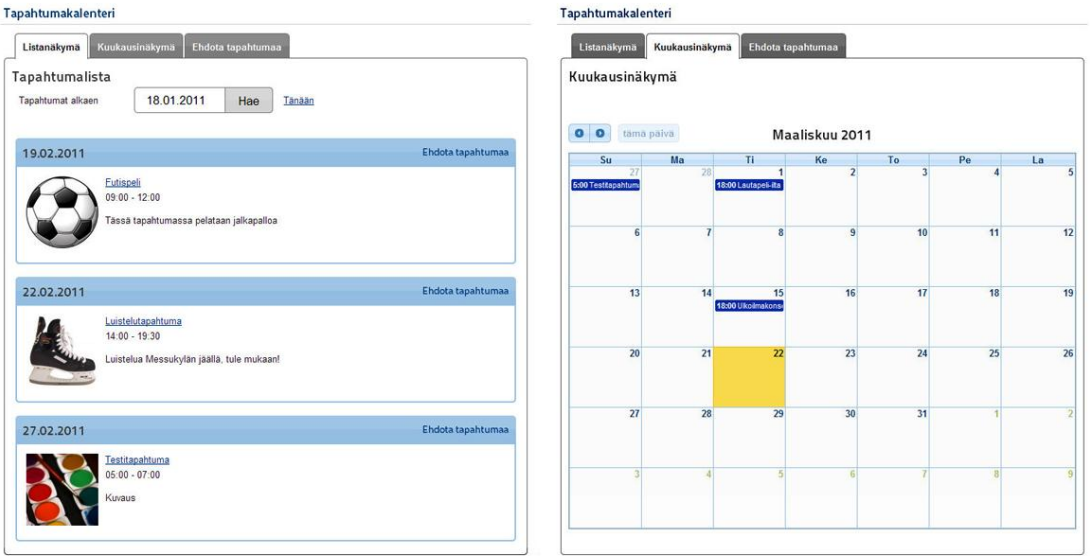
Taulukko 3: Dokumentit.

Event calendar with location support

Overview

Purpose of the project was to design and implement shared event calendaring system for the web sites maintained by Helka ry. Helka stands for Helsingin kaupunginosayhdistys. Helka ry maintains over 40 separate web sites for different districts of Helsinki.

The system is based on Joomla web publishing system and supports geolocation and eventhub feeds. Specific requirement was the functionality to add new calendars to the system and ability to spread the system across to multiple Joomla installations. Technical implementation was divided roughly to two main parts. Calendar interface was implemented as Joomla component and events were saved to shared database. Integration was made for location database supplied by the customer. Calendars support listing events as feeds, and it's planned to use the feeds as content for information displays located in several places in Helsinki.



Screenshots of event list and monthly views of calendar component.

Organization and management

Project team consisted of two project leaders Jami Lehtovirta and Pasi Lampinen. Project staff originally consisted of six people, but two members quit the course. Team members were Jouni Vaaramo, Markus Ijäs, Markus Kumpulainen and Jari Rantanen. Markus Ijäs was representing usability team. Responsibilities varied so that Jouni and Tommi were responsible of coding, servers and database planning, Markus I. and Markus K. were responsible for User interface concepts and usability tests, Jari responded for html coding and end user documentation.

Roles of the project leaders was roughly divided so that Jami focused more for reporting and task planning, and Pasi concentrated more technical aspects of the project.

Weekly meetings were the basis of team work. The progress of the project was tracked and new tasks were given during weekly meetings. After the meetings the memo was saved to shared DropBox. Todo lists were also sent by email. Hours were tracked during the meetings and reported to course leader. Customer had also access to all meeting memos and other project generated material through shared folder.

Methods and tools

Project was implemented as Joomla 1.6 component. Joomla runs on LAMP platform consisting of Linux operating system, Apache web server, MySql database and uses PHP programming language. Various tools and platforms were used for coding, communication and file sharing during project.

1. Joomla 1.6
2. MySQL
3. Google Code
4. SVN (subversion), Tortoise -client
5. DropBox
6. Hour reporting tool
7. Flowdock
8. Present.ly
9. Skype
10. Mockingbird

Joomla 1.6 was not published at the time when project started. Implementation started with release candidate version available. New versions appeared relatively often during project and caused constant need to upgrade development platforms.

Google Code was selected as main tool for the project. Google Code has built in SVN version management, issue tracker and wiki. However task management was lacking. It was possible to use issue tracker for task management also, but this was not so intuitive. Hour reporting tool was implemented by Jami at start of the project. Functionality was based on web form built on Lotus Domino. Tool was usable and it was used during whole project.

Project files and memos were shared between the project members and the customer by using shared DropBox folder. Experiences were positive and usage remained active during project.

Web based Flowdock was used as online chatting and communication tool. The functionality of the tool was quite adequate. However the tool was given up because licensing changed from freeware to monthly fee during the project. Flowdock was replaced with Present.ly. Present.ly was used mainly for notifications and scheduling information. Skype was used for group calls between project leaders and customer.

Mockingbird was used for user interface planning and for usability testing. Mocking bird is web based service with monthly fee, but one of the team member had access to it.

Project phases and development model

Modified Scrum was used as project model. Daily meetings were impossible to arrange in practice, so they were replaced as weekly meetings. Project was divided to preliminary planning phase and four sprints. There was total of four project reviews where customer and course leader were present.

- 4.9.2010 Preliminary planning starts
- 20.9.2010 First meeting with project team
- 14.10.2010 Reviewing of project plan
- 8.10.2010 Requirements
- 18.10.2010 Sprint 1 starts
- 21.10.2010 Requirement spec finalization with customer
- 8.11.2010 Sprint 1 ends, database spec and proof of concepts
- 15.11.2010 Sprint 2 start
- 9.12.2010 Sprint 3 start
- 24.1.2011 Sprint 4 starts, project review,
- 17.2.2011 Project review, usability test results available
- 11.3.2011 Last weekly meeting
- 13.3.2011 Final report submitted
- 14.3.2011 Feedback meeting
- 25.3.2011 Project closing, casual gathering of project team

Experiences

Technology used at the project was relatively new for project members. We prepared for that by studying the Joomla at the beginning of the project. However, some technology related problems were unforeseen, and actually arising outside Joomla. We had some problems with databases and especially with subversion source code management system which was not so straightforward to use that we expected.

Two project members left the project during course including technically highest skilled member, leaving only one full time coder and making implementation phase slow compared to other tasks. The risk for team inactivity or member leaving the project was foreseen as almost all members of the team had to work and study, but it was not clearly thought how to compensate if the risk was actually met. Due the problems, estimated amount of work hours was not reached during the course.

Project was considered success despite the fact that all of the requirements were not implemented. Feedback from the team and customer were extremely positive. Project was liked because of practical hands on doing instead of theory. Interesting aspect was also that project was having somehow problematic past, as it was failed before by commercial software company.

Statistics

Graph of weekly working hours at fig. 1 shows that project was started strongly. Start of the implementation phase was especially slow, but when everything was working speed was increasing after new year. Peaks of the graph somehow correlate with project reviews and other milestones.

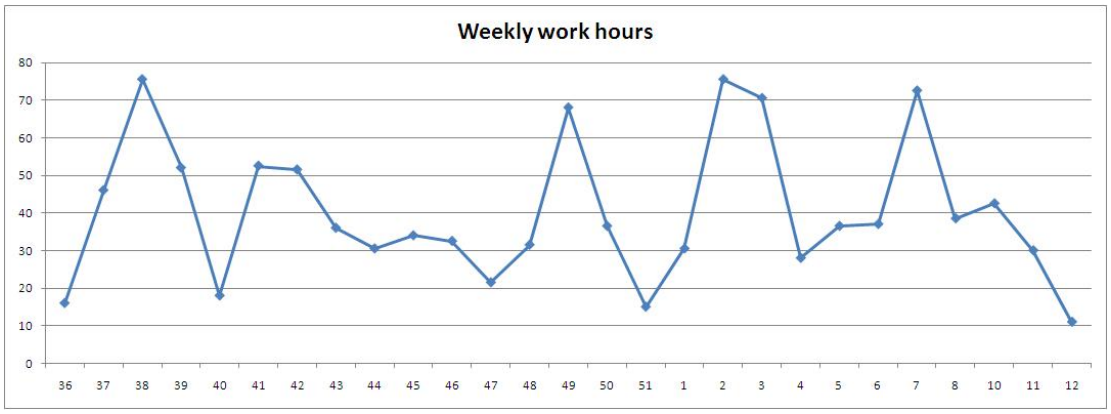


Fig 1, Graph of weekly working hours. X-axis contains number of the week, Y axis contains working hours.

Team size	Dev. model	Start date	End data	Days	Hours
2+5+1	Scrum	4.9.2010	25.3.2011	202	1089,5

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	335	17	12	242,5	4,5	21	7	74	55,5	768,5
%	43,6	2,2	1,6	31,6	0,6	2,7	0,9	9,6	7,2	100%
Usabi-lity	26	31,5	95	2,5	0	6	7,5	61	98,5	321
Total	361	48,5	107	245	4,5	27	7,5	135	154	**1089,5**

Table 2: Group effort by activity.

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
26	6	16	23	1	9

Table 3: Requirements and high-level design outcomes.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
	3	2	0	0	2

Table 4: Design outcomes.

Document	Pages	Versions
Preliminary analysis	12 (slides)	1
Project Plan	28	2
Usability analysis	20	1
Requirements specification	6	1
Design plan	9	1
Test plan	0	0
Test report	0	0
Usability test report	6	1
Final report	33	3
Project's story	6	2
Weekly reports	27 (pcs)	

Table 5: Documents.

Language	PHP	Javascript	CSS	XML	SQL
LOC	4591	7222	3534	64	74
Blank	1709	1148	656	38	15
Comment	1725	1908	896	72	10
Files	82	30	14	2	2

Table 6: Codelines.

Majava3

Overview

Our project group worked on Majava site (www.majava-kilpailu.fi). Majava (Beaver) is an annual Informatics and Computer Science competition for students of different ages.

One goal of the project was to organize Majava 2010 competition in Finland. Other tasks were to add some features, for example practice side of the system in English, remove bugs, redesign admin side of the system and make small changes to the layout. We also created a project video that can be seen at http://www.youtube.com/watch?v=0GT2-kpbXlc.

Organization and management

Project managers: Taina Lempiäinen and Ville Pylkki

Project members: Markku Hintala, Antti Kiiskinen, Karl Ots, Erja Salminen, Paavo Virta and Jenni Paukkunen (usability team)

Client: Timo Poranen represented Department of Computer Sciences

Other stakeholders are Nokia Research Center in Toijala, Finnish National Board of Education and Päivölä.

Even at the end of the project only part of the project group had skills in programming with Ruby. Those who didn't handle Ruby focused for example on the usability and documentation. Approximately three project members focused on coding and three members on other tasks.

Methods and tools

The project was developed with Ruby on Rails framework. PostgreSQL was used for the database and NetBeans IDE for programming. Version control was handled with TortoiseSVN. Oracle VM VirtualBox was used for running the local development environment on the virtual machine. Google Docs was used to collect working hours.

Communication occurred mostly by email and partly by IRC. Documents were stored in Redmine which also was used for task management and host our wiki page.

Most of the tools used worked well. The group however had many problems with Ruby and Ruby on Rails and would have preferred to develope the site with another technology.

Project phases and development model

The development model used was highly modified Scrum. Sprints were about 3 weeks long and there were 8 iterations in the project. The project group met weekly and communicated with e-mail or IRC.

Here are most important dates of the project. Two last iterations didn’t have a review meeting.

Reviews	Date
Preliminary analysis	21.09.2010
Project Plan	12.10.2010
1st review	11.10.2010
2nd review	01.11.2010
3rd review	22.11.2010
4th review	13.12.2010
5th review	19.01.2011
6th review	17.02.2011
Final report	14.03.2011

The project consisted of two big milestones. At the fall our goal was to successfully organize Majava 2010 competition. After that and during the spring we focused on developing new features and redesign the site.

Experiences

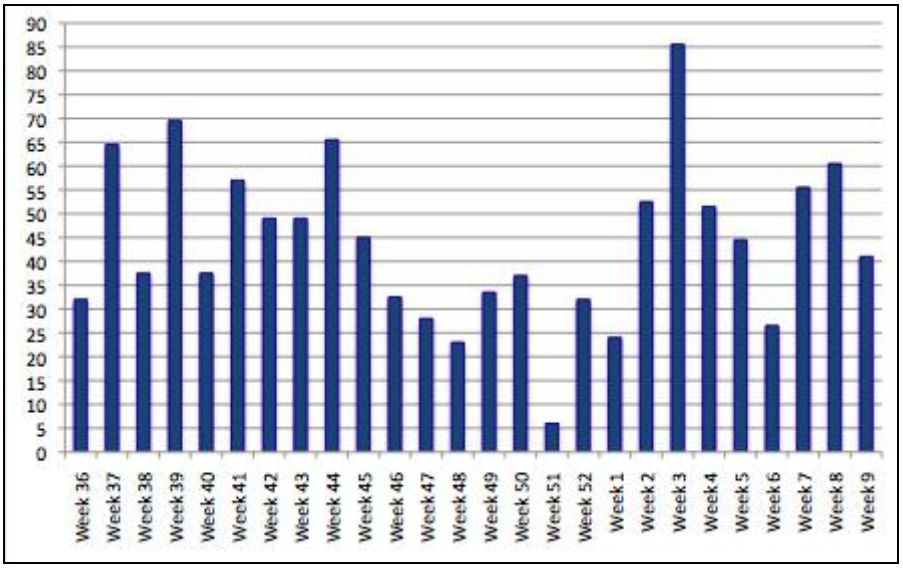
Foreseen risk was that Ruby was a new programming language for almost every member of the group. As was said earlier Ruby was too difficult for part of the project group and so all the members couldn't program. Those who handled Ruby didn't think Ruby as excellent programming language but thought that the whole system should be rebuilt on top of another technology platform.

One unforeseen risk was that updating from Ruby on Rails 2 to Rails 3 caused troubles which caused schedule problems during spring.

Fortunately our hosting service provider arranged flexible performance improvement during the competition. The service provider also arranged recurrent full virtual machine backups.

Statistics

Underneath is a diagram of working hours per week.



Picture 1. Weekly working hours

Team size	Dev. model	Start date	End data	Days	Hours
2+5+1	Scrum	14.9.2010	14.3.2011	-	1205

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	312.5	4.5	61	160.5	96	38.5	34.5	163	269.5	1140
%										
Usabi-lity	21				11			1	32	65
Total										**1205**

Table 2: Group effort by activity.

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
124	-	-	2	1	17

Table 3: Requirements and high-level design outcomes.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams
-	-	-	-	-	-

Table 4: Design outcomes.

Document	Pages	Versions
Preliminary analysis	7	4
Project Plan	21	7
Heuristic evaluation	8	4
Requirements specification	124 Majava3_requirements.pdf	-
Design plan	2	2
Test plan admin site	5	1
Test plan competition site	7	1
Test/usability report admin site	12	5
Test/usability report competition site	19	5
Tietotekniikan Majava-kilpailu	37	Regularly

- vuoden 2009 tehtävät		updated
Tietotekniikan Majava-kilpailu - vuoden 2010 tehtävät	46	Regularly updated
Final report	21	2
Project's story		1
Weekly reports	25	-
Project video	9:39	1
Formal project video	4:19	1

Table 5: Documents.

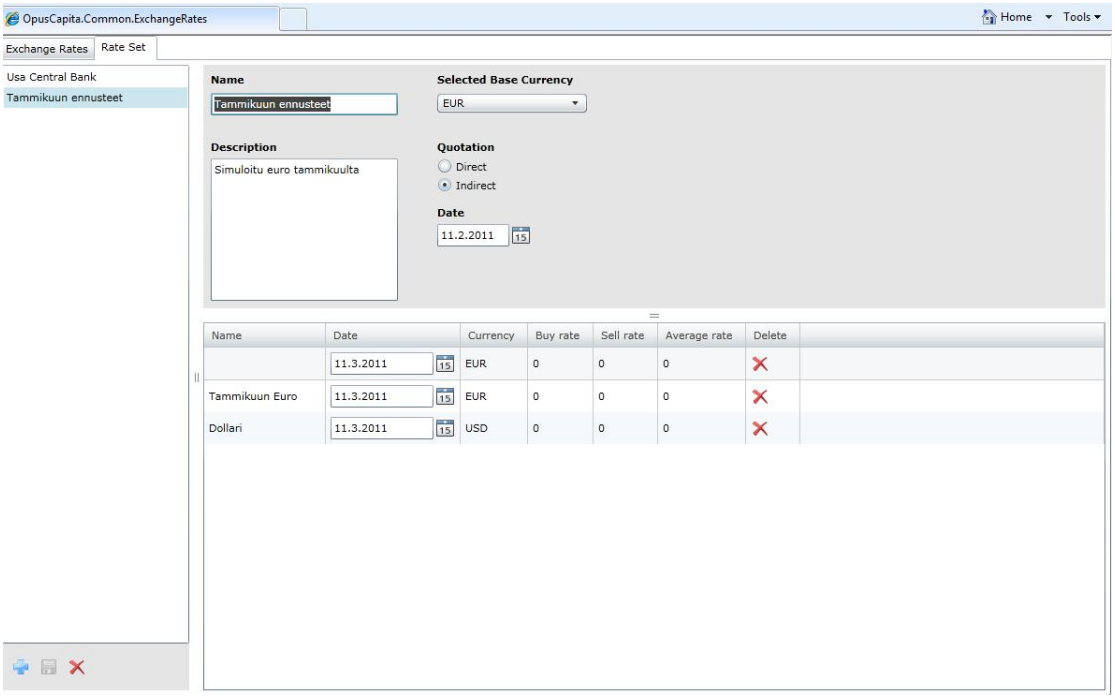
Language	Ruby
LOC	
SLOC	
Reused code	-
Reused and modified	-
Classes	-
Functions	-
Code revisions	347

Table 6: Codelines.

Team OpusCapita

Overview

Users of OpusCapita software family do not yet have the possibility to browse or enter exchange rates using web UI. In the near future web UI will be the only way to use OpusCapita software, which is why it is very important to create totally new web UI component for rate sets to work as a part of OpusCapita software family. The task for Team OpusCapita was to create a portion of the OpusCapita software to enable viewing and creation of currency rates on the new web UI. The further goal of the project was to improve the usability of the function, which was found to be lacking in the previous versions.



Normally the program receives currency rate information through the server from various providers, such as banks, but a further functionality was implemented for the users to create 'rate sets' of their own in order to display specific rate values of their own as needed.

Organization and management

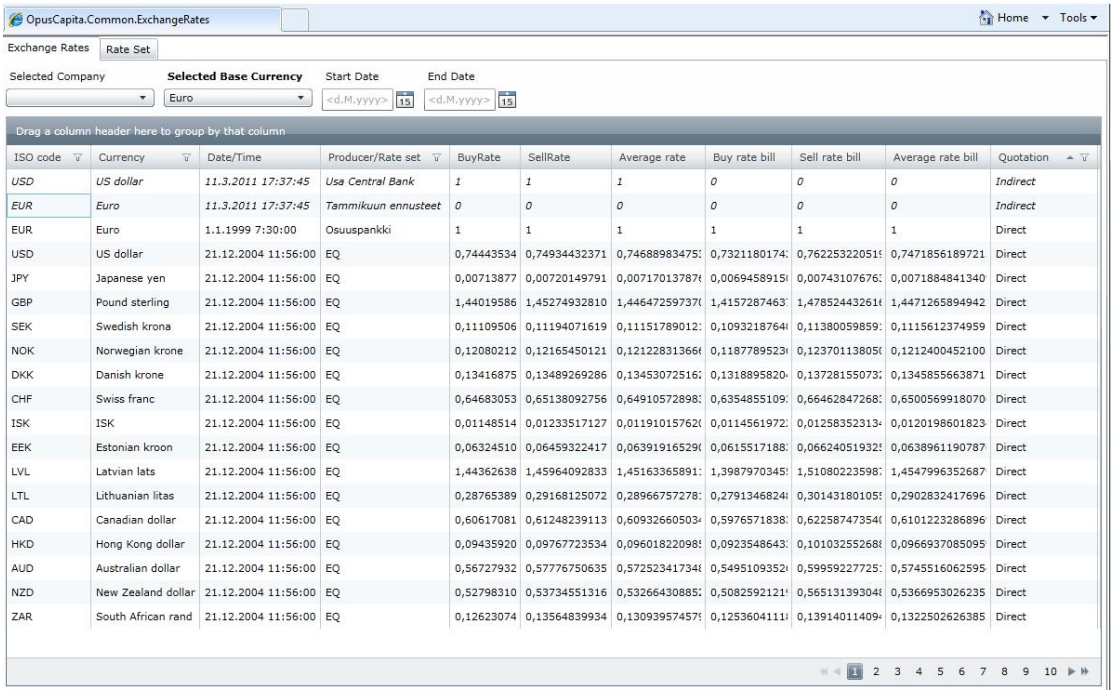
The project was done for OpusCapita oy, with Jouni Kirjola working as the client's contact.

Project members

- Tommi Pirttiniemi (Project Manager)
- Matti Ollila (Project Manager)
- Teemu Jääskeläinen (Project Member)
- Aaro Kortesmaa (Project Member)

- Heikki Kuivala (Project Member)
- Teemu Ruotsalainen (Project Member)
- Minna Vangonen (Project Member)
- Lasse Mäkinen (Project and Usability Team Member)

Methods and tools



ISO code	Currency	Date/Time	Producer/Rate set	BuyRate	SellRate	Average rate	Buy rate bill	Sell rate bill	Average rate bill	Quotation
USD	*US dollar*	*11.3.2011 17:37:45*	*Usa Central Bank*	*1*	*1*	*1*	*0*	*0*	*0*	*Indirect*
EUR	*Euro*	*11.3.2011 17:37:45*	*Tammikuun ennusteet*	*0*	*0*	*0*	*0*	*0*	*0*	*Indirect*
EUR	Euro	1.1.1999 7:30:00	Osuuspankki	1	1	1	1	1	1	Direct
USD	US dollar	21.12.2004 11:56:00	EQ	0,74443534	0,74934432371	0,74688983475	0,7321180174	0,762253220515	0,7471856189721	Direct
JPY	Japanese yen	21.12.2004 11:56:00	EQ	0,00713877	0,00720149791	0,00717013787	0,0069458915	0,00743107676	0,0071884841340	Direct
GBP	Pound sterling	21.12.2004 11:56:00	EQ	1,44019586	1,45274932810	1,44647259737	1,4157287463	1,47852443261	1,4471265894942	Direct
SEK	Swedish krona	21.12.2004 11:56:00	EQ	0,11109506	0,11194071619	0,11151789012	0,1093218764	0,11380059859	0,1115612374959	Direct
NOK	Norwegian krone	21.12.2004 11:56:00	EQ	0,12080212	0,12165450121	0,12122831366	0,1187789523	0,12370113805	0,1212400452100	Direct
DKK	Danish krone	21.12.2004 11:56:00	EQ	0,13416875	0,13489269286	0,13453072516	0,1318895820	0,13728155073	0,1345855663871	Direct
CHF	Swiss franc	21.12.2004 11:56:00	EQ	0,64683053	0,65138092756	0,64910572898	0,6354855109	0,66462847268	0,6500569918070	Direct
ISK	ISK	21.12.2004 11:56:00	EQ	0,01148514	0,01233517127	0,01191015762	0,0114561972	0,01258352313	0,0120198601823	Direct
EEK	Estonian kroon	21.12.2004 11:56:00	EQ	0,06324510	0,06459322417	0,06391916529	0,0615517188	0,06624051932	0,0638961190787	Direct
LVL	Latvian lats	21.12.2004 11:56:00	EQ	1,44362638	1,45964092833	1,45163365891	1,3987970345	1,51080223598	1,4547996352687	Direct
LTL	Lithuanian litas	21.12.2004 11:56:00	EQ	0,28765389	0,29168125072	0,28966757278	0,2791346824	0,30143180105	0,2902832417696	Direct
CAD	Canadian dollar	21.12.2004 11:56:00	EQ	0,60617081	0,61248239113	0,60932660503	0,5976571838	0,62258747354	0,6101223286896	Direct
HKD	Hong Kong dollar	21.12.2004 11:56:00	EQ	0,09435920	0,09767723534	0,09601822098	0,0923548643	0,10103255268	0,0966937085095	Direct
AUD	Australian dollar	21.12.2004 11:56:00	EQ	0,56727932	0,57776750635	0,57252341734	0,5495109352	0,59959227725	0,5745516062595	Direct
NZD	New Zealand dollar	21.12.2004 11:56:00	EQ	0,52798310	0,53734551316	0,53266430885	0,5082592121	0,56513139304	0,5366953026235	Direct
ZAR	South African rand	21.12.2004 11:56:00	EQ	0,12623074	0,13564839934	0,13093957457	0,1253604111	0,13914011409	0,1322502626385	Direct

1 2 3 4 5 6 7 8 9 10

Many tools were used to help to organize and chat about the project. Most used tool was email because everyone used it so it was easy to contact the entire group. IRC was handy tool to get quick answers for instance in coding problems and to have conversations about various issues with other team members altough only half of the team was using it. Meeting times were scheduled through Doodle and some of the common documents were jointly created with Google Docs.

The project was done with Visual Studio 2010, which was used to keep track of the team tasks, share the code and documentation and also for the development of the program itself. Framework used was Silverlight 4, which worked admirably for the project. The main languages used were XAML and C# and the user interface components were to large part done with Microsoft Blend.

Architectural pattern used was Microsoft's own MVVM which bears a heavy resemblance to the well known MVC -model. It took some time to learn, but it became increasingly clear as the project advanced and visualizing it turned out to be a good move for clarity.

Project phases and development model

The project was done in Agile manner, using a modified version of SCRUM heavily altered to fit both the requirements of the project course, and the relatively small

ammount of working hours team members had allocated for the course per week. Especially daily meetings were handled during weekly meetings, supplemented by conversations by mail and IRC.

A large issue with the project was that there were no real specifications for the user interface, and thus a major part of the project was to actually design it from scratch and incrementally update it based on its usability and feedback from customer meetings. A major reason for this approach was the customer's refusal to show the original functionality, with the statement that any impression from the original functionality would likely be a negative one. This approach in turn blurred the lines between the development stages, with semi-functional software being developed early on in the project with most of the time being spent on repeated iterations on improving the usability.

The project was divided into various sprints with somewhat vaque goals, in addition to the constant re-iteration to improve usability. Following are the major points in the course timeline.

Timeline:

Initiation (Sprint 0), wk 36 - 38

- Gathering up the team
- Setting up environments
- First look at the customer requirements
- Getting to know the tools

Preliminary analysis meeting 23.9.

Initial design (Sprint 0), wk 39 - 41

- First prototype
- Meeting client for clarification on the requirements and showing the first prototype design

First customer meeting, 18.10

Sprint 1, wk 41 - 43

- Proper sprints started
- First proper framework implementation of the program

Sprint 2, wk 44 - 46

- Simple functionality to display data implemented
- MVVM workshop
- Blend workshop

- Team Repository workshop

Customer GUI meeting, 29.11

Sprint 3, wk 47-49
- First UI design revision based on customer feedback
- Usability/UI workshop
- Data filtering

Sprint 4, wk 2 - 4
- Complex functionality to display data implemented

First course review meeting, 14.1

Customer GUI meeting. 26.1

Sprint 5, wk 5 - 7
- Data input functionality
- Adding last missing functionality

Second course review meeting, 10.2

Sprint 6, wk 8-10
- Final touches on the code
- Creating project CD

Final course meeting, 13.3

Experiences

Due to either expert preparation, or exceptional luck, the team only encountered two major problems, both of which had been expected:

Busy team members - Many members of the team had day jobs as well, and most had otherwise busy schedules. The major problems caused by this were the uncertainty that certain tasks would be done at a specific time, problems arranging live meetings and lack of meetings where the entire team would be together at once. These problems were managed with services that allowed voting on meeting times (such as Doodle), and updates by email and irc to the missing team members. This obviously interfered

with proper SCRUM practices, but there may not be any better sollution to this problem when the course work is somewhat low priority compared to members' other engagements.

Abiguity - The customer requirements were hazily defined, some of the functionality did not exist in previous versions and the customer forbade us of studying the previous versions as starting point. Furthermore, both the industry in question and the tools we were required to use were new to the team members. This mainly required large amounts of studying time and multiple iterations of design and customer meetings to clear out. For future preparations further stressing the ways of coping with this would possibly help: Training tasks, workshops and squeezing out as much information out of the customer as humanly possible and as early as possible.

Statistics

Team size	Dev. model	Start date	End date	Days	Hours
2+5+1	SCRUM	11.9.2010	11.3.2011	183	1168

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
TeamTotal	262,5	27	36,5	167	29	9,5	4	310,5	185	1031
Team %	25%	3%	4%	16%	3%	1%	~0%	30%	18%	100%
Usa. Total	41,5		21	9	9			23,5	33	137
Usa. %	30%	0%	15%	7%	7%	0%	0%	17%	24%	100%
Total	304	27	57,5	176	38	9,5	4	334	218	**1168**

Table 2: Group effort by activity.

Document	Pages	Versions
rautalankamalli_valuuttakurssienasettamisen	1	1
rautalankamalli_valuuttakurssienselaaminen	1	1
OC - UI hahmotelma - Tiedon selaaminen - Tuottaja	1	1

OC - UI hahmotelma - Tiedon selaaminen - Valuutta kurssit	1	1
OC - UI hahmotelma - Tiedon syöttäminen	1	1
Vaatimusmaarittely_ValuuttakurssienSelailu	9	7
Vaatimusmaarittely_ValuuttakurssienSyotto	9	7
Testaustapauksia	18	8
Tasklist	2	1
Architecture	7	1
Project Story		1
opus-capita-Hanke-ehdotus	2	1
Preliminary analysis	3	1
Project plan	18	1
Final report	TBA	TBA

Table 3: Documents.

Language	C#, XAML
LOC	1340
SLOC	1340
Reused code	0
Reused and modified	0
Classes	25
Functions	34
Code revisions	-

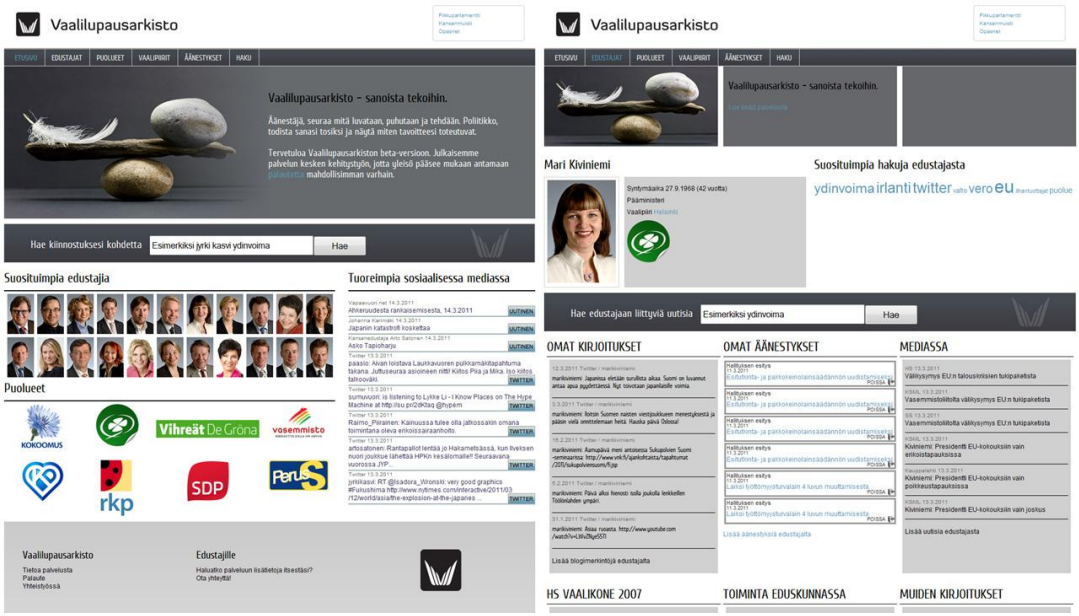
Table 4: Codelines.

Vaalilupausarkisto

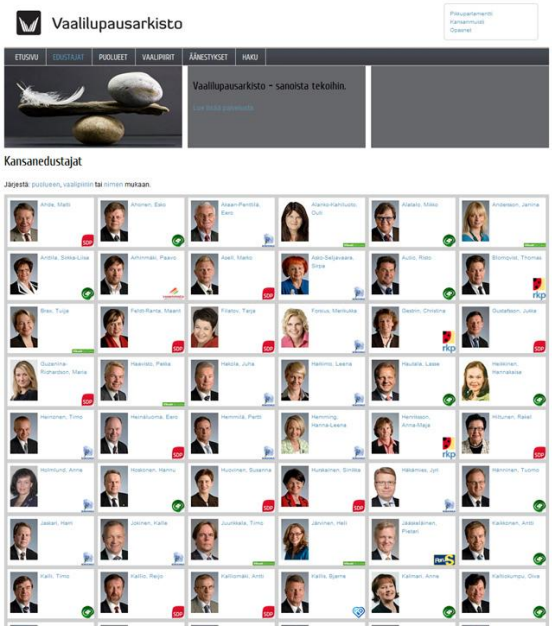
Overview

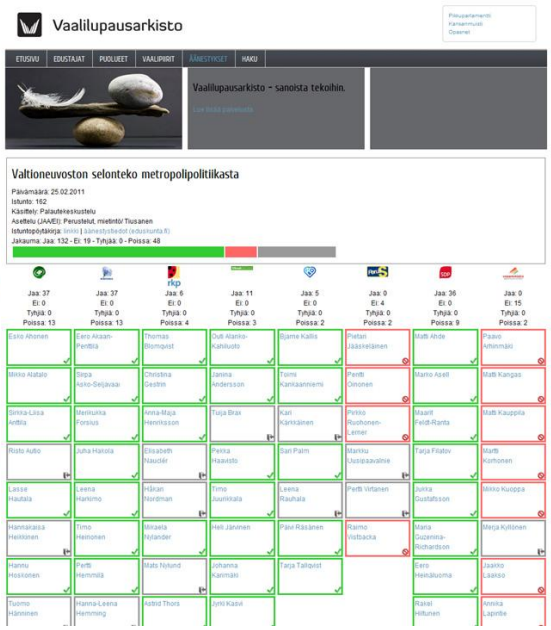
"Election promise archive" is a web site for political memory. You can find politicians' words, deeds and their news from the site. Politicians can find what is written about them and they can see their own action seen in public. Voters can see what is said and what is done.

In other words our group designed and implemented a web-portal providing information (=blogs, news, votes, etc) about the parties and members of the Finnish parliament gathered from public RSS feeds and other information sources. The main idea is to show all the data considering defined keyword in one place.



The portal was made from scratch and after making the basic functionality we started gathering data sources and designed how data should be shown. The layout was implemented before the beta and before that we concentrated on functionality.





Organization and management

Project Managers: Jukka Pollari (Project management, Redmine updates, testing)
Antti Syrjä (Project management, server administration, UI)

Members: Jouni Kähkönen (coding, search engine)
Janne Redsven (coding, database)
Mika Kähkönen (coding, data sources)
Johan Laitinen (coding, UI)
Jussi Kivinen (testing, documenting)
Jenni Paukkunen (UI team)

Client: Raimo Muurinen, politics student.

Group members had good basics for PHP coding so the language was easy to choose.

Methods and tools

Selected tools were very good. We used Subversion for the code repository. Everyone used different GUIs for browsing the svn depending on their operating system. Also the coding environment was chosen for own habits without any line. The project data was in Redmine. We used Redmine for issues and hour keeping and the we had our meeting memos and wiki in Redmine. For the documents we user Google Docs because of the good collaboration chances.

We decided to use CakePHP framework for our code and it was a good decision. CakePHP saved tens of hours because we didn't have to do all the groundwork.

Project phases and development model

We used applied SCRUM. We had one weekly meeting and after that we used IRC and email to keep up almost daily with the issues. Project managers were much online so it was easy to contact other if you had to ask something.

We had meeting every Monday 16-17.30. Member presented what they how done during the week. Then we organized tasks for the next week. Project managers had their own meeting before group to see what has to be done and checked. Meetings were regular and efficient and the attendance was very good.

We had two bigger checkpoints; demo version done before 24.12.2010 and beta version to be published 14.2.2011 at Nettikansa meeting at Uusi Tehdas.

Task	**Date**
Start meeting	13.9.2010
Preliminary analysis	23.9.2010
Project plan	3.10.2010
Sprint I	4.10- 10.10.2010
Sprint II	11-24.10.2010
Sprint II Review	25.10.2010
Sprint III	25.10-7.11.2010
Personal report I	1-7.11.2010
Sprint III Review	8.11.2010
Sprint IV	8-21.11.2010
Sprint IV Review with Timo Poranen	22.11.2010
Sprint V	22.11.-6.12.2010
Project presentation	1.12.2010
Sprint V Review	7.12.2010
Sprint VI	7-16.12.2010
Sprint VI Review	16.12.2010
Sprint VII	17.12.2010- 9.1.2011
Sprint VII Review	10.1.2011

Personal report II	1.1.-16.1.2011
Sprint VIII	10.1.-23.1.2011
Sprint VIII Review with Timo Poranen	24.1.2011
Sprintti IX	24.1.-6.2.2011
Sprintti X	7.2.-20.2.2011
Usability tests	10.2.2011
Beta launch	14.2.2011
Sprintti X Review	21.2.2010
Project prestentation	25.2.2011
Final report	6.3.2011
Feedback meeting	7.3.2011
Project story	11.3.2011
Project CD	11.3.2011
Personal report III	March 2011

Experiences

The group members commitment to project was very good and collaboration worked like a dream. Group members had good coding abilities and we didn't have much technical problems.

Of course we had some challenges. The requirements were sometimes a bit roundabout and we had to modify customers thought and specifications a bit. Some features come too late in the course. We should have freezed our backlog. Most of the important (and original) requirements were implemented as planned. In the end we noticed that we scheduled some requirements too high and some too low. But it was partly because of the new requirements. Because of the good group commitment the 240 hour limit came too early for some members.

Cooperation with external suppliers (graphics, Finnish parliament) caused some extra

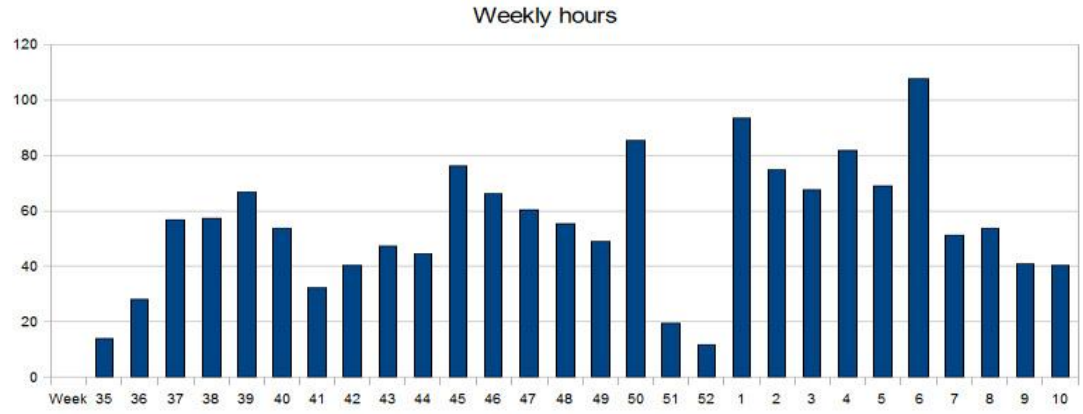
work because we had to done them ourselves before we got the real ones.

Other things to mention

Vaalilupausarkisto ry association was founded during the spring. Client and three group members are involved. Also a Protomo project will be established. The INFIM department has a project for next summer to extend the story search.

So our work with Vaalilupausarkisto will be continued by these projects

Statistics



Hours for members

Antti Syrjä	241.25
Jukka Pollari	243.75
Jouni Kähkönen	249.25
Janne Redsven	190.75
Mika Kähkönen	247.25
Johan Laitinen	240.25
Jussi Kivinen	137.75
Jenni Paukkunen	102.50

Team size	Dev. model	Start date	End data	Days	Hours
2+5+1	SCRUM	11.10.10	13.03.11	152	1656.5

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	391.5	115.5	89	312.5	111.25	163	136.5	182	155.25	1656.5
%	23,63	6,97	5,37	18,87	6,72	9,84	8,24	10,99	9,37	100%
Usabi-lity	37,5	38	15	-	10,5	-	-	1,5	-	102.5
										1759

Table 2: Group effort by activity.

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
100	-	5	~ 10 (psd)	6	34

Table 3: Requirements and high-level design outcomes.

Pages	Overview diagrams	Class diagrams	Sequence diagrams	State diagrams	Other diagrams

Table 4: Design outcomes.

Document	Pages	Versions
Preliminary analysis	5	
Project Plan	22	Updated constantly
Usability analysis	7	5
Requirements specification	100 issues in RM	
Design plan	-	
User interface document	-	
Test plan	-	
Test report	-	
Usability test plan	16	
Usability test report	18	8

Final report	26	19 “updates”
Project's story		1
Weekly reports	24	

Table 5: Documents.

Language	PHP
LOC	73900 lines in our app folder
SLOC	-
Reused code	CakePHP has plenty of lines
Reused and modified	-
Classes	-
Functions	-
Code revisions	-

Table 6: Codelines.

Luuppi-projekti

Yleistä

Projektin lopullisena tuotoksena syntyi uusittu www-sivusto Luuppi ry:lle. Luuppi ry on Tampereen yliopiston matematiikan, tilastotieteen ja tietojenkäsittelytieteiden opiskelijoiden ainejärjestö. Projektissa uudistettiin portaalin tekninen toteutus, toiminnot sekä visuaalinen ilme. Sivusto siirtyy lähitulevaisuudessa kokonaisuudessaan Luuppi ry:n hallintaan, jonka jälkeen se tulee esille osoitteeseen www.luuppi.fi. Sivusto toimii yleisimmillä selaimilla.

Yleisilme uuden sivuston etusivulta. Oikealla näkyvissä hallinnointipalkki, joka näkyy kun käyttäjällä on admin-oikeudet (Kuva 1).

Kuva 1

Uuden staattisen sivun lisääminen onnistuu nykyään myös selaimen kautta, kunhan käyttäjällä on vaaditut oikeudet. Kuvassa on erillinen tila englanninkieliselle sisällölle suomalaisen sisällön vieressä(Kuva 2).

Kuva 2

Organisaatio ja projektinhallinta

Projektin asiakas Luuppi ry on Tampereen yliopiston matematiikan, tilastotieteen ja tietojenkäsittelytieteiden opiskelijoiden ainejärjestö.

Asiakkaan edustaja:

Tuomas Tauriala

Projektipäällikköt:

Panu Tunttunen

Petri Ikävalko

Projektiryhmänjäsenet:

Mikko Kuivanen

Johannes Lampela

Eero Jaakonaho

Kari Jussila

Käytettävyysryhmän jäsen:

Saila Oldén

Menetelmät ja työkalut

Projektin kotisivuina toimivat projektisivut osoitteessa http://code.google.com/p/luuppi/, jossa pidettiin yllä lähes kaikkea projektiin liittyvää informaatiota. Tapaamisten pöytäkirjat, linkit työkaluihin, vinkit, dokumentit ja työtuntien seuranta olivat kaikki samassa portaalissa. Projektisivusto toimi pääosin moitteettomasti, työkaluna siinä ei ollut moittimista. Kommunikointiin ryhmä käytti aluksi omaa irc-kanavaa, mutta sen käyttö laantui hieman projektin edetessä. Intensiivisten koodausjaksojen aikana irc-kanava oli erittäin toimiva ratkaisu ja muina aikoina sähköposti oli helpoin ja tavoittavin viestintäväline ryhmän kesken. Toisinaan ryhmällä tosin oli vaikeuksia vastata lähetettyihin viesteihin ja itsekuri oman raportoinnin suhteen lipsui.

Toteutuskieleksi valittiin PHP ja IDE:nä käytettiin netBeansia, jossa oli käytössä myös iden oma versionhallinta. Projektissa käytettiin myös Codeigniter kehitysympäristöä ja javascript-kirjasto JQueryä.

Projektin vaiheet ja kehitysmalli

Projektissa käytettiin SCRUM-ohjelmistokehitysmallia, jossa pyrittiin noin kolmen viikon sprintteihin. Toisinaan sprinttien aikataulut joustivat, mutta jokainen sprintti pyrittiin lopettamaan ennen uusia tehtäviä. Loppua kohden tehtävien hajanaisuus kuitenkin hieman hajoitti sprinttien rakennetta, eikä tarkkoja sprinttikohtaisia tehtäviä pystynyt erittelemään kunnolla. Tämä tapahtui kuitenkin pääasiassa vasta seitsemännen sprintin aikana/jälkeen. Sprinttienaikaiset tapaamiset ja keskustelutuokiot irc-kanavalla toimivat ryhmän sisäisinä ohjausmekanismeina.

Projektin vaiheet sprinttien avulla ilmaistuna, jokainen sprintti alkoi työnjaolla ja valmistuneet osat kasattiin yhteen sprintin lopuksi.

Sprintti	**Viikot**
Sprintti 0	38-40
Sprintti 1	41-43
Sprintti 2	44-46
Sprintti 3	47-49
Loma	50,51
Sprintti 5	1-3
Sprintti 6	4-6
Sprintti 7	7-

Taulukko 1: Sprintit

Seuraavassa on listattuna projektin tärkeimmät virstanpylväät, eli milestonet. Näissä vaiheissa projektia raotettiin muille ryhmille ja asiakkaalle, sekä arvioitiin tilannetta.

Tapahtumat	Pvm.
Esitutkimus	vk38
Projektisuunnitelma	vk40
Katselmointi	7.10.2010
Katselmointi	10.11.2010
Projektiesitys	1.12.2010
Loppuesitys	25.02.2011

Taulukko 2: Projektin milestonet

Johtopäätökset

Tarkka odotettujen riskien luettelu löytyy projektin projektisuunnitelmasta. Seuraavassa eriteltynä projektin kohtaamat riskit.

Projektiryhmäläisen lopettaminen: Yksi ryhmämme jäsenistä jätti kurssin kesken melko aikaisessa vaiheessa. Tästä ei ollut vielä niin paljoa haittaa, sillä projektille jäi yhä 4 ohjelmoijaa.

Projektiryhmäläisen pitkä loma: Yhdessä yhden lopettaneen ryhmäläisen kanssa tämä aiheutti jo hieman hankaluutta ryhmän toiminnalle, eikä projektin edetessä pystytty jakamaan työtaakkaa yhtä tasaisesti kaikkien harteille.

Väärät arviot kyvyistä: Projektiryhmä uskoi omiin kykyihinsä aluksi hieman liikaa ja toteutettavien osien määrä yhdessä vaaditun opiskelun kanssa osoittautui turhan suureksi. Tekniikat ja toteutettavien osien määrä saatiin kuitenkin hallintaan projektin edetessä.

Motivaation puute: Joissakin projektin vaiheissa ryhmäläisten motivaatio tuntui rakoilevan, eikä annettuja tehtäviä saatu valmiiksi aikataulussa.

Kokemuksen projekti oli opettavainen ja osoitti miten vaikea on saada paperilla hienosti järjestetyt asiat toimimaan käytännössä samalla tavalla. Paljon kuitenkin tuli opittua projektin aikana ja luultavasti seuraava projekti sujuu jokaiselta ryhmäläiseltä jo huomattavasti paremmin. Tärkein vaihe projektin onnistumisessa varmasti on aloitus, jonka aikana ryhmä saadaan sitoutettua projektiin, tai suhtautumaan siihen epäillen. Kannattaakin muistaa jokaista projektia aloittaessa ottaa huomioon jokaisen osallistujan motivointi ja sitouttaminen, tämä onnistui vaihtelevasti omassa projektissamme. Projektista jäi hyvät opit myös projektinhallintavälineiden ja kehitysympäristön käytöstä. Edellämainittuja saa ilmaiseksi käyttöön ja niiden hyödyntäminen täysipainoisesti helpottaa projektissa tarvittavien rutiinien suorittamista.

Tilastot

6.1 Projektin yleiset tiedot

Ryhmän koko	Kehitysmalli	Aloituspäivä	Lopetuspäivä	Päiviä yht.	Tunteja yht.
2+4+1	SCRUM	10.9.10	13.5.11	253	1118

Taulukko 3: Projektin tiedot

6.2 Projektiryhmän työtunnit

Taulukko 4: Projektiryhmän kokonaistunnit

Toiminta	Projektin suunnittelu ja johtaminen	Vaatimusten määrittely	Suunnittelu	Ohjelmointi	Integrointi ja testaus	Katselmoinnit	Korjaus	Opiskelu	Muut	Yht.
Tunnit	278	23	88	302	10	31	81	186	119	1118
%	25	2	8	27	1	3	7	17	11	100%

6.3 Dokumentaatiot

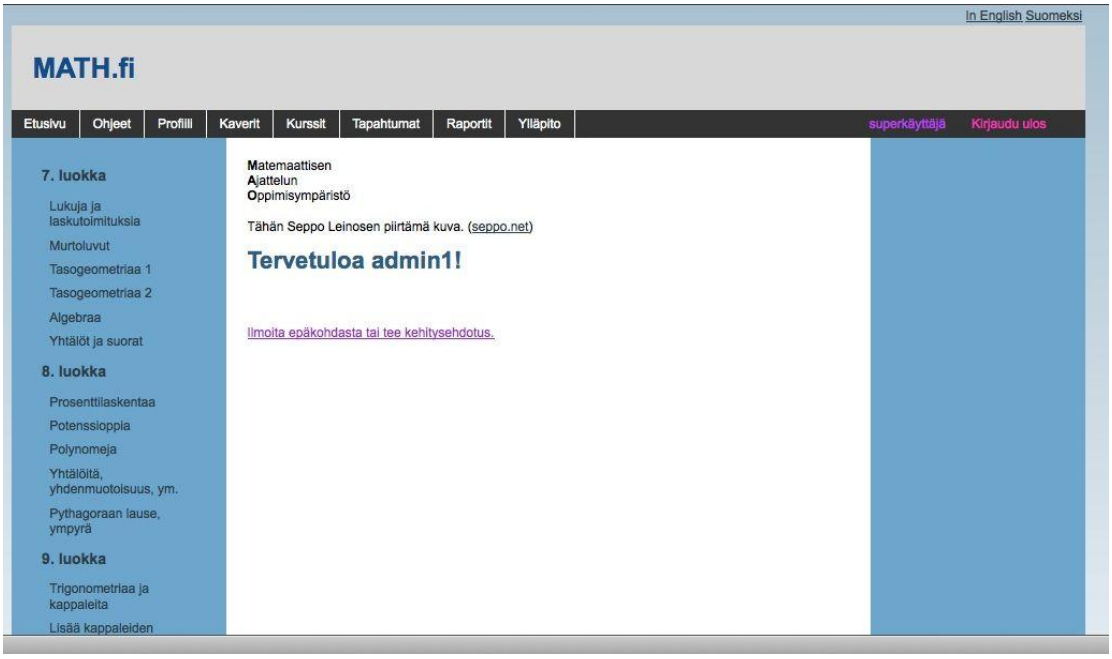
Dokumentti	**Sivut**	**Versio**
Esitutkimus	10	1.0
Projektisuunnitelma	19	1.2
Vaatimusmäärittely	23	1.2
Testaussuunnitelma	25	1.0
Projektikertomus	5	1.0
Viikkoraportit	16	-

Taulukko 5: Dokumentit

Matemaattinen oppimisympäristö

Yleiskuvaus

Toteutimme projektityönä uudelleen matemaattisen oppimisympäristön (matematiikka.net) . Kyseessä on matemaattisen ajattelun oppimisympäristö perusopetuksen 7. - 9. luokille. Tuote pyrkii tieto- ja viestintätekniikan avulla ohjaamaan oppilaita itseohjautuvaan opiskeluun. Ensisijaisesti tuotetta tulee käyttämään Oriveden yhteiskoulu, mutta jatkossa mahdollisesti levikki laajenee. Tuote ei ole vielä käytössä tietokannasta puuttuvan sisällön vuoksi.



Kuva 1: Aloitussivu

Organisointi ja hallinta

Projektiryhmä koostui kahdesta projektipäälliköstä ja kuudesta projektityöntekijästä.

Managereina toimivat:

- Anne Mikkonen
- Mikko Rantanen

Projektityöntekijöitä olivat:

- Pasi Kiema (vaatimusmäärittely ja koodi)
- Hanne-Lotta Mäenpää (tietokanta, käyttöliittymä ja koodi)
- Jukka Springare (käyttöliittymä)
- Tahvo Repo (tietokanta ja koodi)
- Tero Strakh (käyttöliittymä ja koodi)
- Saana Riihelä (käytettävyysasiantuntija)

Projektipäälliköiden työt jakaantuivat siten, että Anne hoiti dokumentointia, piti yhteyttä kurssin vetäjään Timo Poraseen sekä veti katselmointeja. Mikko puolestaan oli vastuussa viikkopalavereista ja -raporttien lähettämisestä sekä toteutusryhmän opastuksesta.

Menetelmät ja työkalut

Kehitysympäristö perustettiin asiakkaan webhotellipalveluun. Verkkopalvelimena toimii Apache 2.x. Tietokanta toteutettiin MySQL 5 -tietokantahallintajärjestelmällä. Ohjelmointityökaluna käytettiin Netbeans -ohjelmaa, johon integroitiin svn-versionhallinta.

Ohjelmointikielenä oli PHP 5. Tämän lisäksi käytettiin ajax:ia ja javascript:a. Käyttöliittymät toteutettiin XHTML -kielellä ja niiden ulkoasut on muokattu CSS -tyyliohjeiden mukaan. Ohjelmointiympäristöksi valittiin CakePHP 1.3, mikä tukee MVC-mallia.

Vaatimuksia hallittiin Pivotal Tracker -työkalulla. Sillä oli helppo listata vaatimuksia, mutta sprinttien hallinta oli vaikeaa, varsinkin kuin ne eivät toteutuneet samanpituisina. Sekä managerit että ryhmäläiset kokivat työkalun käytön hankalaksi. Pivotal Tracker on jatkossa muuttumassa maksulliseksi, joten sitä tuskin tullaan valitsemaan jatkossa opiskeluprojekteihin.

Projektiin liittyvät dokumentit ja asiat kerättiin ryhmän projektiwikiin. Tietokantakaaviot piirrettiin Dia:lla. Muu dokumentointi hoidettiin pääasiallisesti Microsoftin word -ohjelmalla ja ne käännettiin pdf-muotoon.

Tuntikirjanpito tehtiin Timelet -ohjelmalla. Varsinkin toteutusryhmä käytti yhteydenpitoon omaa irc-kanavaa.

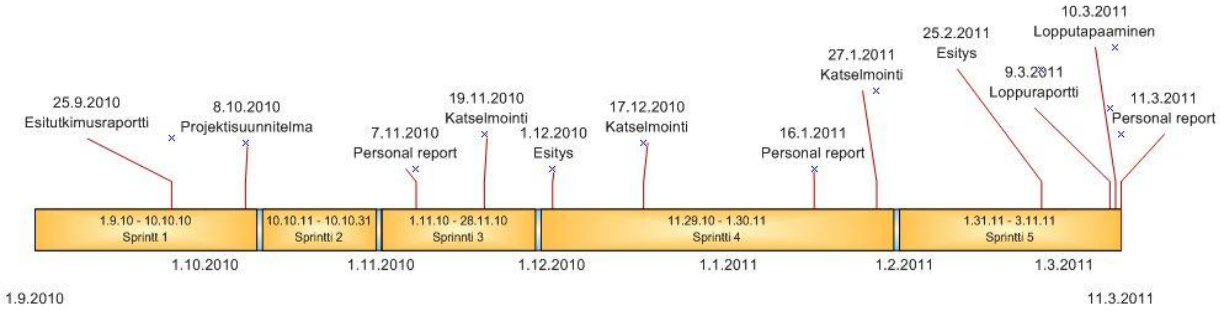
Projektin eteneminen kehitysmalli

Kehitysmenetelmänä käytettiin ketterää menetelmää Scrum:a. Aluksi projekti jaettiin kuuteen sprinttiin, mutta venyneen joulutauon takia neljäs ja viides sprintti yhdistettiin. Päivittäisten scrum-kokousten sijaan pidettiin palaveri viikoittain, maanantaisin. Sen lisäksi lähes joka torstai tavattiin irc -kanavalla. Tapaamisissa käytiin läpi edellisen viikon tekemisiä ja sovittiin tulevasta. Joskus saatettiin jäädä tapaamisen jälkeen pienemmällä porukalla ratkomaan jotakin ongelmaa, tai ryhmäläiset pitivät omia esim. käyttöliittymäsuunnittelutyöryhmiä.

Menetelmää ei saatu kovin hyvin toimimaan. CakePHP vaati paljon opiskelua ja osa toteutettavista asioista jäivät kesken sprintin aikana. Arviomme siitä, kuinka kauan jonkin tekemiseen menisi, ei pitänyt paikkaansa. Lopulta oli oikeastaan mahdotonta hallinnoida projektia sprinteittäin ja päädyimme vain priorisoimaan asioita sitä mukaa, kun toteutus eteni. Toteutusjärjestystä käytiin läpi aina viikkopalavereissa ja irc:n välityksellä. Iterointi vaatii todella paljon työtä, jos projekti ei pysy alusta asti jollain lailla aikataulussa.

Hyvä puoli on se, että heti alkuun projekti tuli suunniteltua aikataulullisesti kokonaan, vaikka se ei toteutunutkaan. Kokonaisnäkemys asioista auttoi kuitenkin hahmottamaan sitä, mitä jatkossa vaaditaan ja mitä osioita ei missään nimessä ehditä tekemään.

Aikataulu ja tärkeimmät päivämäärät projektin osalta olivat:



Kuva 2: Projektin aikataulu.

Kokemuksia

Projektityökurssi on hyvin opettavainen kurssi. Monia työkaluja käytettiin ensimmäistä kertaa. Kurssin työläydestä johtuen, se koetaan joskus jopa rankkana. Omien taitojen kanssa saa painia jatkuvasti, mutta toisaalta se opettaa oman ajankäytön suunnittelua ja ongelmanratkaisukykyä. Lisäksi tiimityön tekeminen tuo hyvää harjoitusta työelämää varten. Projektin aikana pystyy yhdistelemään eri kursseilla oppimiaan asioita.

Projektin aihe oli kiinnostava ja se motivoi toteuttajia. Myös asiakkaan aktiivinen osallistuminen koettiin positiivisena asiana. Ryhmän henki oli loistava. Apua annettiin aktiivisesti sitä kaipaaville. Mielialaa kuitenkin laski se, ettei kaikki vaikuttaneet motivoituneelta osallistumaan.

Vaikka projektissa ei saatu toteutettua kaikkea mitä suunniteltiin, oli se jokseenkin onnistunut. Alkuun päästiin aika myöhään siitä syystä, että alustan valinta tehtiin niin myöhään. Toisaalta parempi tehdä perustan valinta perusteellisesti, kun hätiköiden. Myös ajoittain työkiireet ja muut opinnot veivät niin paljon aikaa, että projekti eteni hitaasti. Toteuttajaryhmän yhteistyö sujui kiitettävän hyvin ja yleisilmapiiri oli rento. Se mihin olisi kaivannut enemmän panostusta oli käyttöliittymäsuunnittelu ja ulkoasun viimeistely. Valitettavasti käyttöliittymäpuoli ei ollut ihan niin sitoutuneita asiaan kuin olisi toivonut.

Vaatimusten hallinnassa olisi ollut parantamisen varaa. Vaatimusmäärittelyt olisi pitänyt pystyä järjestämään paremmin ja seurata miten ne valmistuvat. Mutta siksi opiskellaan, että osataan huomioida nämä asiat oikeassa työelämässä.

Tilastot

Team size	Dev. model	Start date	End data	Days	Hours
2+5+1	Scrum	17.9.2010	11.3.2011	-	1194,5

Talukko 1: Projektin tiedot.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	422	9,5	120	290,5	6	29	8	225	84,5	1194,5
%	35,32	0,8	10,05	24,32	0,52	2,41	0,67	18,84	7,08	100%

Taulukko 2: Ryhmän tunnit.

Number of requirements	Pages	Käyttötapauksia	Näyttöjä	Tietokantakaaviot	Tietokantataulut
34				2	33

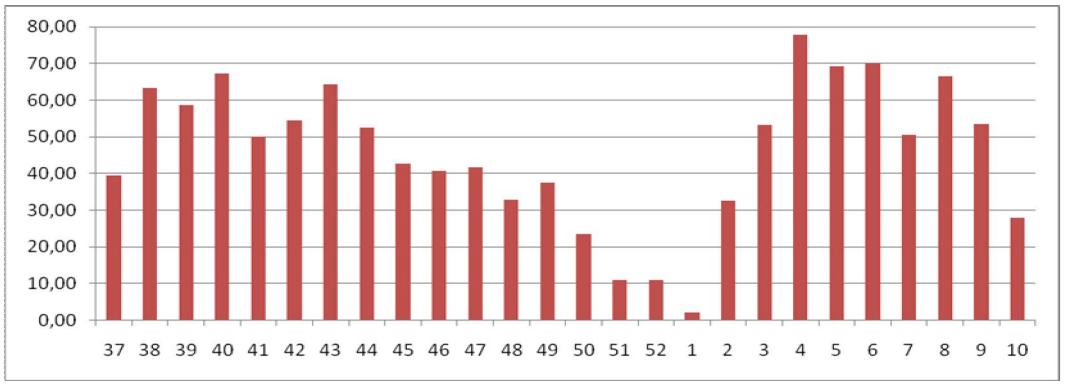
Taulukko 3: Suunnittelu.

Document	Pages	Versions
Preliminary analysis	7	1.1
Project Plan	21	1.7
Usability analysis	14	1.7
Test plan	6	1.0
Final report	12	1.1
Project's story	5	
Weekly reports	20	

Taulukko 4: Dokumentit.

Language	PHP	Javascript	CSS
LOC	3106	621	871

Taulukko 5: Koodirivit



Kuva 3: Projektin viikkotunnit.

MoTiPe

Yleistä

Projekti toteutti 3D-grafiikkaan ja uusiin syöte- ja tulostuslaitteisiin pohjautuvan tilallisen pelin. Pelin ohjauksessa käytettyjä syötetapoja ovat sensorit ja blobo-pallot. Pelin grafiikan lisäksi käytetään muita elementtejä kuten tilaääniä ja valoja. Peli toimii noin 5x5m kokoisessa tilassa. Pelin kohderyhmänä ovat kouluikäiset lapset ja nuoret.

Organisaatio ja projektinhallinta

Projektin asiakas oli Tampereen yliopiston tietojenkäsittelytieteiden laitoksen yksikkö TAUCHI.

Asiakkaan edustajat:

- Markku Turunen
- Jaakko Hakulinen

Projektipäällikköt:

- Rami Saarinen
- Pauli Lammi

Projektiryhmän jäsenet:

- Hanna Heinänen
- Jonne Iso-Tuisku
- Sebastian de Mel
- Ville Saarinen
- Aleksi Tiensuu

Käytettävyysryhmän jäsen:

- Minna Hara

Menetelmät ja työkalut

Toteutuksen alustana käytettiin Panda3D pelimoottoria. Toteutus tehtiin Python ohjelmointikielellä.

Lähdekoodin versionhallintaan käytettiin Subversionia(SVN). Ryhmä käytti pääsääntöisesti TortoiseSVN-ohjelmaa.

Projektin hallinnan ohjelmistona käytettiin VersioOne-ohjelmaa joka koettiin hieman kankeana ohjelmana käyttää.

Ryhmän eniten käytetty työkalu kurssin aikana oli DokuWiki, jonka avulla seurattiin ryhmän työtunteja ja koottiin dokumentaatio yhteen paikkaan.

Yhteydenpidossa käytettiin sähköpostilistaa sekä virtuaalipalavereissa Jabber viestintämenetelmää. Jabberin käyttö lopetettiin projektin loppuvaiheessa, koska siitä

saatu hyöty koettiin vähäiseksi.

Ryhmä käytti epävirallisena viestintävälineenä IRC-kanavaa. Tämän käyttö koettiin hyödyllisenä reaaliaikaisena viestimistapana. IRC-kanavaa käytti noin puolet ryhmäläisistä.

Projektissa hyödynnettiin useita erilaisia syöte- ja tulostelaitteita, kuten

- Blobo-palloja 2kpl, jotka ovat langattomia peliohjaimia
- Kaikusensoreita, jotka määrittelevät suurimman kohteen etäisyyden kaikuluotauksen avulla
- Lasersensoreita, jotka ilmoittavat kun lasersäde katkeaa lähettimen ja vastaanottimen välillä.
- MIDI tiedonsiirtojärjestelmä, joka mahdollistaa erilaisten sähkölaitteiden välisen kommunikoinnin.

Projektin vaiheet ja kehitysmalli

Projektissa käytettiin Scrum -kehitysmallia. Kehitysmalli ei kuitenkaan kovin hyvin sopinut projektiin, sillä mallia ei saatu kunnolla otettua käyttöön. On ilmeistä, että kehitysmalli ei kovin hyvin soveltunut projektiryhmälle. Haasteellisesta alusta lähtien ryhmän oli vaikea omaksua rutiininomaista käyttötapaa projektin työkaluille. Eritoten tämä näkyi puutteellisena tehtävien tuntiarvioiden merkitsemisenä. Tuntiarvioiden tekeminen on keskeistä Scrum -mallissa, koska niiden avulla luodaan ja pidetään yllä ns. burndown kuvaajaa, mikä on koko prosessin tärkein mittari.

Myöhemmin vuodenvaihteessa projektin johto muutti projektimallia siten, että yksilönvapautta tehtävien valinnan suhteen rajoitettiin ja projektipäälliköt alkoivat aktiivisemmin nimeämään henkilöitä tehtävien tekijöiksi. Tämähän kulkee täysin kaikkea sitä vastaan, mitä Scrum:n pitäisi olla: ryhmäläisen tulisi olla asiantuntija, jolla on vapaus ja vastuu valita ja määritellä tehtäviä sprintin kulun aikana. Tämän asiantuntijan tärkeimmät ominaisuudet ovat itseohjautuvuus, aktiivisuus ja yhteystyökyky. Oikein toimivassa Scrum -tiimissä ei välttämättä edes tarvita ScrumMasteria, koska ryhmä osaa mukautua tilanteeseen nopeasti ja saumattomasti.

Samaan aikaan kun projektimallia vietiin autoratiivisempaan suuntaan, osittain luovuttiin VersionOnen käytöstä sprinttien ylläpitämiseen ja siirryttiin pitämään sprinttitiedot wikissä. Tämä oli seurausta juuri siitä, että ryhmän oli vaikea omaksua työkalun käyttö ja täten se jätettiin taustalle.

Projektin vaiheet: 1. Tutkiskeluvaihe ilman tehtävänantoa

Projektin aluksi emme saaneet varsinaista tehtävänantoa asiakkaalta. Tämä johti siihen, että siirryimme tutkimus ja kartoitusvaiheeseen tehtävänantoa odotellessa. Vaiheen prioriteettina oli opiskella Pythonia ja Panda3D:tä, vertailla muita mahdollisuuksia ja laatia muutamia demoja, joiden haluttiin ruokkivan asiakkaan mielikuvitusta ja mielenkiintoa.

Vaiheen artifakteina voidaan pitää kartoitusta 3D pelimoottoreista ja kahta Python/Panda3D demoa.

Projektin vaiheet: 2. Tehtävänanto saatu, käytettävä laitteisto ja tila hankittavana ja remontissa

Saatuaan tehtävänannon asiakkaalta ryhmä alkoi suunnitella pelille alustavaa arkkitehtuuria ja pelin tarinaa. Tuotanto ei tässä vaiheessa päässyt juuri käyntiin, koska ryhmällä ei ollut saatavilla tarvittavia laitteita tai rajapintoja.

Projektin vaiheet: 3. Blobot ja äänet saadaan käyttöön

Aluksi ryhmä sai käytettäväkseen kaksi Blobo-palloa ja täten pelillisten interaktioiden suunnittelu saattoi alkaa. Asiakkaalla oli myös työn alla rajapinta Blobojen käyttöön. Ryhmä kuitenkin huomasi rajapinnan olevan liian rajoittava ryhmän toimintaan ja täten Blobot integroitiin järjestelmään omilla ratkaisuilla.

Tässä vaiheessa ryhmä sai myös äänirajapinnat käyttöönsä. Myöhemmin äänirajapinta korvattiin Panda3D:n omalla ratkaisulla.

Projektin vaiheet: 4. Valot ja tila saadaan käyttöön

Marras- Joulukuussa ryhmä pääsi kehittämään sovellusta varsinaisessa tilassa. Ensimmäinen varsinainen pelidemo valmistui ja loi pohjan jatkokehitykselle. Sensoreita ei tässä vaiheessa vielä ollut. Valojärjestelmä otettiin käyttöön vaiheen loppupuolella.

Projektin vaiheet: 5. Sensorit saadaan käyttöön

Viimeinen varsinainen kehitysvaihe. Tässä vaiheessa saatiin myös sensorit auttavasti mukaan, vaikka niiden sijoittelu tilassa olikin vielä kesken. Sensorit integroitiin järjestelmään ja valoja hyödynnettiin kaikissa pelin osa-alueissa.

Projektin vaiheet: 6. Koodauksen alasajo, dokumentointi ja testaus

Tässä vaiheessa tuotantoa alettiin ajaa alas ja pääpaino siirrettiin dokumentointiin ja testaukseen. Muutamaa poikkeusta lukuunottamatta tässä vaiheessa sai vain tehdä virhekorjauksia.

Katselmoinnin aihe	**Katselmointiajankohta**	**Projektin vaihe**
Esitutkimus	29.9.2010	vaihe 1 ja 2
Projektisuunnitelma	22.10.2010	vaihe 2
Katselmointi	3.12.2010	vaihe 2,3 ja 4
Katselmointi	19.1.2011	vaihe 5
Katselmointi	9.2.2011	vaihe 6
Projektikertomus	24.2.2011	vaihe 6
Loppuraportti	24.2.2011	vaihe 6
Projekti-CD		Projektin jälkeen
Lopputapaaminen	24.2.2011	vaihe 6

Taulukko 1: Projektin aikataulu

Johtopäätökset

Ennen kuin projektissa valitaan käytettävä kehitysmalli olisi hyvä saada kunnollinen käsitys esimerkiksi siitä mitä ryhmässä osataan. Hyvä osaaminen vahvistaa omatoimisuutta, joka tukee Scrum -kehitysmallin käyttöä. Projektin aikana oli tehtävä kurssin vaatimuksien mukaiset dokumentaatiot, mitkä eivät ole Scrum -dokumentaatioiden mukaisia. Tämä ehkä toi oman sekaannuksen heille, jotka eivät

Scrum -kehitysmallia vielä tunteneet.

Projektin luonne tuki vahvasti Scrum -kehitysmallin käyttöä, mikä käy ilmi kappaleessa 4. kuvatussa projektin eri vaiheissa. Projektin johdon on kuitenkin hyvä osata reagoida olemassa oleviin tilanteisiin jotta projekti etenisi ja projektin tavoitteet saataisiin tehtyä ajallaan. Projektin edetessä projektin johto mukauti tilanteeseen eri tavoin: projektimallia muutettin siten, että tehtävien tekemiset nimettiin. Projektin käytössä olevien työkalujen pääpainoa siirrettiin ryhmälle helpompaan suuntaan, jolloin viikottaiset tehtävät laitettiin wikiin. Myöhemmin virtuaali-istunnoista, joiden anti ei ollut kovin hyödyllistä projektille, luovuttiin ja siirryttiin sähköpostiraportointiin. Nämä muutokset tuntuivat vaikuttavan positiivisesti projektin etenemiseen ja ryhmän aktiivisuuteen.

Projektin lopputuloksena saimme kursin osalta vaadittavat dokumentaatiot sekä muita lisädokumentaatioita ja toimivan pelin.

Tilastot

Ryhmän koko	Kehitysmalli	Aloitus pvm	Lopetus pvm	tunnit
2+7+1	Scrum	13.09.2010	24.02.2011	1104

Taulukko 2: Projektin tiedot

Proposed	Approved	Implemented	Verified	Deleted
21	19	19	19	2

Taulukko 3: Vaatimukset

Toiminta	Projektin suunnittelu ja johtaminen	Vaatimusten määrittely	Suunnittelu	Ohjelmointi	Integrointi ja testaus	Katselmoinnit	Korjaus	Opiskelu	Muut	Yhteensä
Tunnit	326,5	21	78	238,5	53,5	15,5	5,5	157,5	100,5	996,5
%	32,8	2,1	7,8	23,9	5,3	1,7	0,5	15,8	10	100
Käytettävyys	-	-	-	-	-	-	-	-	-	107,5
Yhteensä	326,5	21	78	238,5	53,5	15,5	5,5	157,5	100,5	1104

Taulukko 4: Projektiryhmän kokonaistunnit

Dokumentti	Sivut	Versiot
Esitutkimus	4	1
Projektisuunnitelma	11	1

Vaatimusmäärittely	7	1
Käytettävyystestin testisuunnitelma	4	1
pelitarina final version	3	1
3D Pelimoottorien vertailu	3	1
Suunnittelu	19	1
Testaussuunnitelma	8	1
Testausraportti	2	1
Loppuraportti	9	1
Projektikertomus	4	1
Viikkoraportit	1	23

Taulukko 5: Dokumentaatiot

OhOp

Overview

The project's goal was to plan and implement a learning environment for programming. System was implemented as an online-service so that users won't have to install development environments or anything else on their own computers. On the website students can read theory of programming and then do programming exercises using only a web browser. The site is planned and implemented so that it can be used to host programming contests and it also can be used in programming classes in schools and universities.

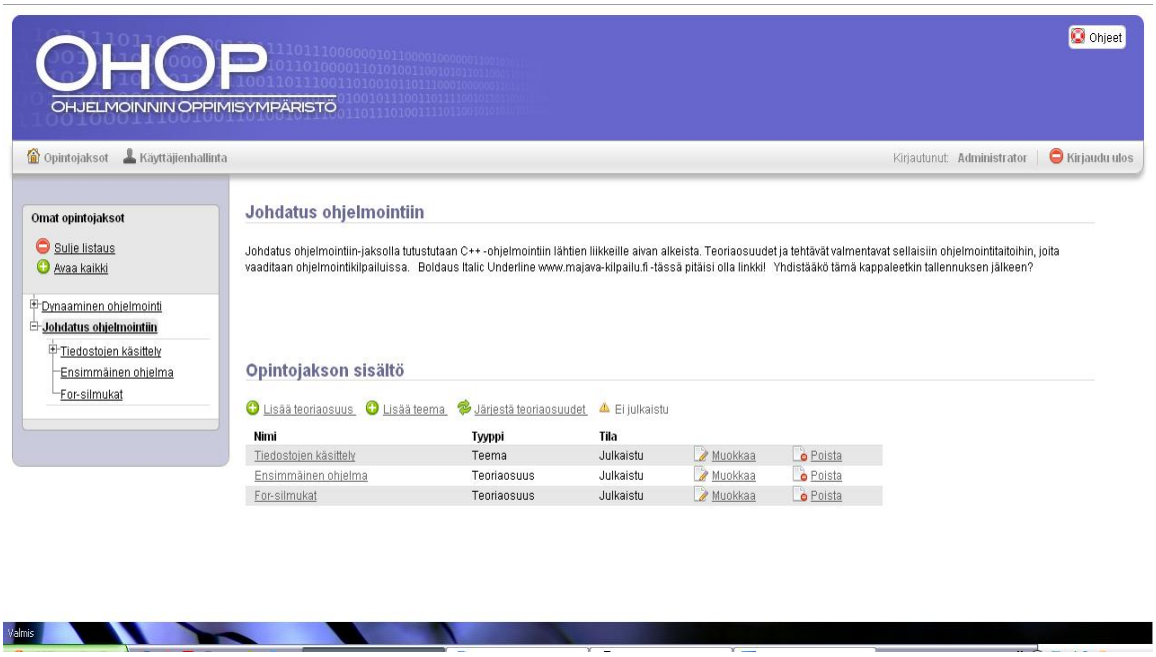


Figure 1: View of course and its content.

Organization and management

Originally the team consisted of eight members: project leaders Johanna Aittoniemi and Kristian Mattila, programmers Aku Häsänen, Nina Leivo, Jani Matkala, Matti Nieminen and Jaakko Routamaa and usability expert Niina Hakala. During the fall two members quit the project work course.

Figure 2: OhOp team. From left to right: Jaakko Routamaa, Johanna Aittoniemi, Kristian Mattila, Niina Hakala ja Jani Matkala. Missing from the picture is Matti Nieminen.

Methods and tools

We used Eclipse and Subversion for development work, and the combination worked really well. The Subversion repository was configured as SVN+SSH with public key authentication in our development server. Eclipse was chosen, because using SVN+SSH with just plain text editors or other IDEs might have been a bit troublesome.

For database management we used PHPMyAdmin which was perfectly suited for the task. We also had another nifty tool in our toolbox called WebSVN, which enabled repository exploring through a web interface.

For testing the code before committing it to the repository, we had testing directories for each developer. A custom PHP script was created for resetting a testing directory to an up-to-date version of the current development branch.

Project phases and development model

OhOp project was implanted by using modified Scrum as its development model. Major difference to regular Scrum was that we didn't held daily meetings, only weekly meetings. The project was divided into a start period, a specification period and seven actual sprints.

The start period lasted approximately 4 weeks; it started 1.9.2010 and ended 26.9.2010. During this time project team met the first time, team consulted the client, project was presented to other project leaders and preliminary analysis was written and reviewed 27.9.2010.

The specification period lasted 2 weeks; it started 27.9.2010 and ended 10.10.2010. During this period, team did a requirements analysis and wrote the project plan, which was reviewed 15.10.2010.

1st sprint lasted 4 weeks; it started 11.10.2010 and ended 7.11.2010. During this period team planned database and user interface and studied Joomla!. Also a development server was set up. No review with clients was held. One member quit the project work course.

2nd sprint lasted 3 weeks; it started 8.11.2010 and ended 28.11.2010. Programming began. Prototype of the user interface was made. One member quit the project work course. Review was held 29.11.2010 with client.

3rd sprint lasted 3 weeks; it started 29.11.2010 and ended 19.12.2010. First version of adding and modifying course was implemented. Coding for the compiler part of the program was started. Review was held 17.12.2010 with clients.

4th sprint lasted 3 weeks; it started 20.12.2010 and ended 9.1.2011. During this time we had a period break and Christmas holidays, so much wasn't done. Review was held 20.1.2011 with client.

5th sprint lasted 3 weeks; it started 10.1.2011 and ended 30.1.2011. During this period team coded course, theory and user management components and compiler, planned usability tests and modified program's layout. Test plan documents were written. Review was held 3.2.2011 with client.

6th sprint lasted 2 weeks; it started 31.1.2011 and ended 9.2.2011. The programming continued. Team started implement exercise component. A review with the client wasn't hold.

7th sprint supposed to last 2 weeks; start 10.2.2011 and end 28.2.2011. Unfortunately project was late so 7th sprint ended 5.4.2011. During the 7th sprint there was a review 24.2.2011, final presentation 25.2.2011, final meeting 5.4.2011 and writing several documents including final report and project story. During this period team also continued coding and did the usability testing for the web site.

Experiences

I worked in the project as a usability expert. Working in a project that lasted over six months developed my patience and ability to plan my actions. The amount of work required increased during the project because two members of our group quitted at the beginning of the executing period. The division of tasks inside the group worked out well and most of the things required were well taken care of, and on time. Personally I lost my motivation when the project did not end on time and I had to concentrate on my other courses as well. The amount of work as an usability expert is less than other members have and I had to do some extra work the Uteam project leaders gave me to reach the required working hours.

-Niina Hakala

I was excited when the project started because the subject of the project sounded interesting. Unfortunately, the project needed much more effort than I expected. In addition to this, we started implementation quite late and two of our group members quit the course. My job constantly required most of my time so my contribution to the

project was pretty small. In addition to that I realized that just maybe the Scrum was not the perfect development model for our team mostly because of the communication with the customer and group members was not continuous. Working with the team was a nice experience and I believe that for people who have not yet worked with a customer this course has a lot to give.

-Matti Nieminen

I had no experience in web developing or project working before this course. I started WWW-programming course at the same time with the project work course to complete my lacking knowledge in WWW-programming. Estimating own resources while considering occasionally strict timetable was one of the biggest challenges of the project. Sometimes I chose too ambitious goals which caused some unnecessary work. This course gave me lots of new experience that will be valuable in upcoming projects.

-Jaakko Routamaa

Project managers gave me pretty much free hands on layout and design issues. I had plenty of space to experiment with my ideas and that kind of a work style suited me well. The work load however didn't spread out evenly: For first two months I didn't really have anything to do and when the things finally started happening it felt like I didn't have time to finalize everything as well as I would have liked. All in all, Kristian, Johanna, Jaakko, Niina and Matti were just nice people to work with, it was a fun project to work on and I really enjoyed doing it!

-Jani Matkala

Project's subject was very interesting from the start and I enjoyed planning the program. There was much work and quite lot of stress, but I also learned much. Team was great and I enjoyed working with them.

-Johanna Aittoniemi

I worked in the project as a project manager, though I spent most of my time planning, coding and configuring the development server. The project was a nice opportunity to try project management in a safe way, without risking actual money. I'm quite satisfied with the outcome and I learnt a lot about using Scrum and how not to use it. All in all, our team worked well together!

-Kristian Mattila

Statistics

Team size	Dev. model	Start date	End data	Days	Hours
2+3+1	Scrum	1.9.2010	5.4.2011	217	1297,92

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	379,25	14,75	88,5	471,92	7,25	34,75	0	157,25	50,75	1204,42
%	31,5	1,2	7,3	39,2	0,6	2,9	0	13,1	4,2	100%
Usabi-lity	44,75	0	32,25	0	6,5	0	0	10	0	93,5
Total	424	14,75	120,75	471,92	13,75	34,75	0	167,25	50,75	**1297,92**

Table 2: Group effort by activity.

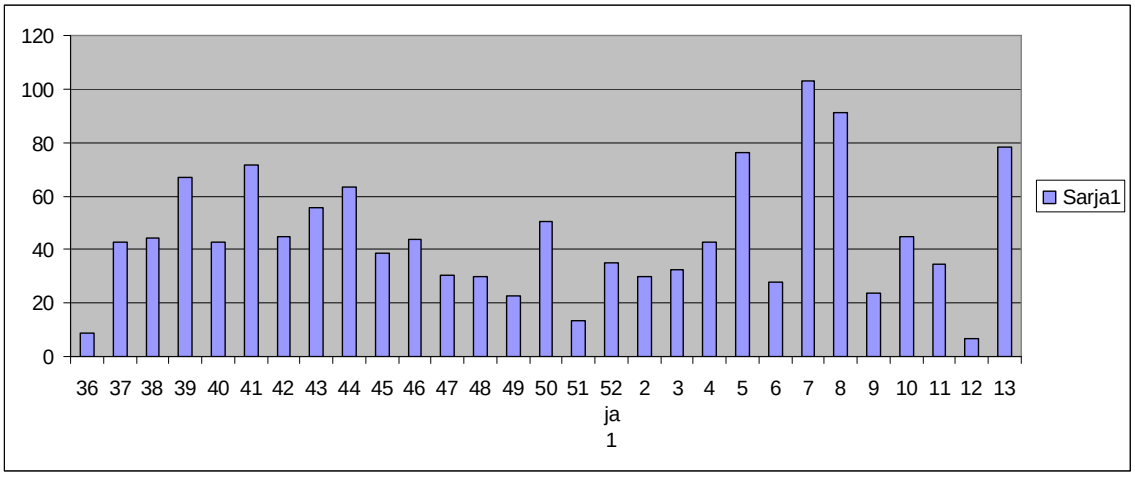


Table 3: Weekly hours.

Number of requirements	Pages	Use-cases	UI screens	Database diagrams	Database tables
85		6	15	1	18

Table 4: Requirements and high-level design outcomes.

Document	Pages	Versions
Preliminary analysis	6	2
Project Plan	14	2
Usability analysis	14	3
User interface document	21	3
Usability Test plan	8	6
Test plan	17	2
Usability test report	19	2

Final report	12	2
Project's story	6	1
Weekly reports	29	

Table 5: Documents.

Language	Files	Blank	Comment	Code
PHP	107	1602	2547	6031
CSS	6	333	51	1431
HTML	3	61	2	419
Javascript	8	71	47	360
XML	11	0	0	355
SQL	2	1	6	8
Sum:	137	2068	2653	8604

Table 6: Codelines.

Unreal Input Testing

Overview

The purpose of the project was to create an input device testing enviroment integrated in to a first person shooter game. The client for this project was Poika Isokoski from TAUCHI -unit in Tampere University

As the rapid development of game industry, FPS games have become one of the most popular game styles. Meanwhile there are many ways of controlling the game character in an FPS game, such as keyboard mouse combination, gamepad etc. However, there is a limited number of open testing environments for implementing new input devices. For this purpose, we are developing this testing environment for easily integrating a new input device to the test enviroment.

The testing environment will provide the basic functionality such as movement (moving forward and backward, turning left and right, etc.), shooting, weapons change, etc. The Unreal Development Kit (UDK) contains a demo game inside which is used as the test enviroment game. Using UDK provides with the possibility to integrate the enviroment to other games created with the same development kit. UDK is free to download for anybody. In addition, the testing environment contains a logging facility in order to estimate the efficiency and accuracy of different input devices.

Organization and management

Team Members:

Kimmo Röppanen (Manager)
Xiaozhou Li (Manager)

Markku Aalto (Developer)
Yunxiang Gao (Developer)
Erkki Heikkonen (Developer)
Wuping Yao (Developer)
Jussi Palomaki (Usability)

Project Management:

The project was mainly managed through weekly meetings and using email and instant messaging (Google wave) for communication on more urgent matters. At the end of each major iteration we had a meeting with the client to show him what was the situation of the project and provide him the possibility for feedback.

Methods and tools

For development we used C# in Visual Studio 2010 with .NET framework for the server side development. On the UDK side the language used was UnrealScript and UDKFrontend was used for compiling. The communication between the game side and server side was handled with a TCP connection.

For Tobii eye tracker functionality testing we had the possibility to use a testing laboratory in Tampere University department of computers sciences.

No real user tests were performed since the product of the project has no clear interface for a user to test but is more a tool for providing developers and researchers an interface to develop and test new input devices.

Project phases and development model

Iteration was used as the development model for this project. There were 6 Iterations all together and at the end of each major Iteration there was a meeting with the client to give him the possibility to give feedback and see the progress of the project.

Iterations:

1st iteration: Learning development kit and relevant knowledge, development plan settled down.

2nd iteration: Develop first runnable test enviroment.

3rd Iteration: Integrate one input device to the game

4th Iteration: Testing & Documentation

5th Iteration: More input devices compliance try out and portability

6th Iteration: Bug fixing, documentations testing

3rd Iteration review meeting documentation:

Review meeting with client 02.12.2010

The review meeting consisted of the client testing the interface with Tobii and giving us feedback of his experience.

observations & suggestions:

small lag in the controlls: eye movement feels slower than in penguin game. Is not necessarily a problem with the interface. (Send timestamps through server when the input is received from tobii to measure the lag for the interface part.)

Separate cursor for showing looking location in-game.

Think about HUD implementation.

Create 'deadzone' in the middle of the screen.

Work on implementing the Logging & analysis tool. (Most important)

Documentation on how to start the game. Developers Document your Code. (Most important)

Other ways to integrate to the game. Create sample classes for integrating through another TCP-server.

Look into creating a build of the modified game so that it can be started from the main menu.

5th Iteration review meeting documentation:

Review meeting 25.01 17-19:

Attendees:

Kimmo Röppänen	PW
Xiaozhou Li	PW
Markku Aalto	PM
Yunxiang Gao	PM
Wuping Yao	PM
Poika Isokoski	Client

-We went throught the current situation of the project and the client seemed happy with the progress.

-Most important now is to get the user manual finished so the client can read it and test it.

-Finish the open issues. (HUD, Timestamp, etc.).

-Agreed to have a last meeting where we should have a finished solution to show.

Experiences

Since half of the team was chinese and half finnish we needed to pay more attention to communication between team members to make sure there were no missunderstandings. Still in the beginning part of our team members ended up spending a lot of time doing some work that was later scrapped from the project.

One of the team member left quite early in the beginning of the project but that was an expected result and didn't really affect much in the project.

Next time we should spend more time on planning and requirements specification and writing everything down more clearly. Now we didn't really do much documentation during the project and this probably made things a little unorganized. But all the team members seemed to be able to cope with it very well. Also we should have planned our testing better which no ended up being almost nonexistent.

Statistics

Team size	Dev. model	Start date	End date	Days	Hours
2+4+1	Iteration	01.10.2010	11.03.2011	162	1100?

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	330	1	41	243	23	16	0	306	44	1004
%	32.9%	0.1%	4.1%	24.2%	2.3%	1.6%	0%	30.5%	4.4%	100%
Usabi-lity	14								4	18
Total	344	1	41	243	23	16	0	306	44	**1022**

Table 2: Group effort by activity

Document	Pages	Versions
Preliminary analysis	6	2
Project Plan	29	4
Requirements specification		
Final report		
Project's story	6	
Weekly reports	20	1

Table 3: Documents.

Language	C#, UnrealScript
LOC	3368
Reused code	126
Reused and modified	126
Classes	22
Functions	144
Code revisions	99

Table 4: Codelines.

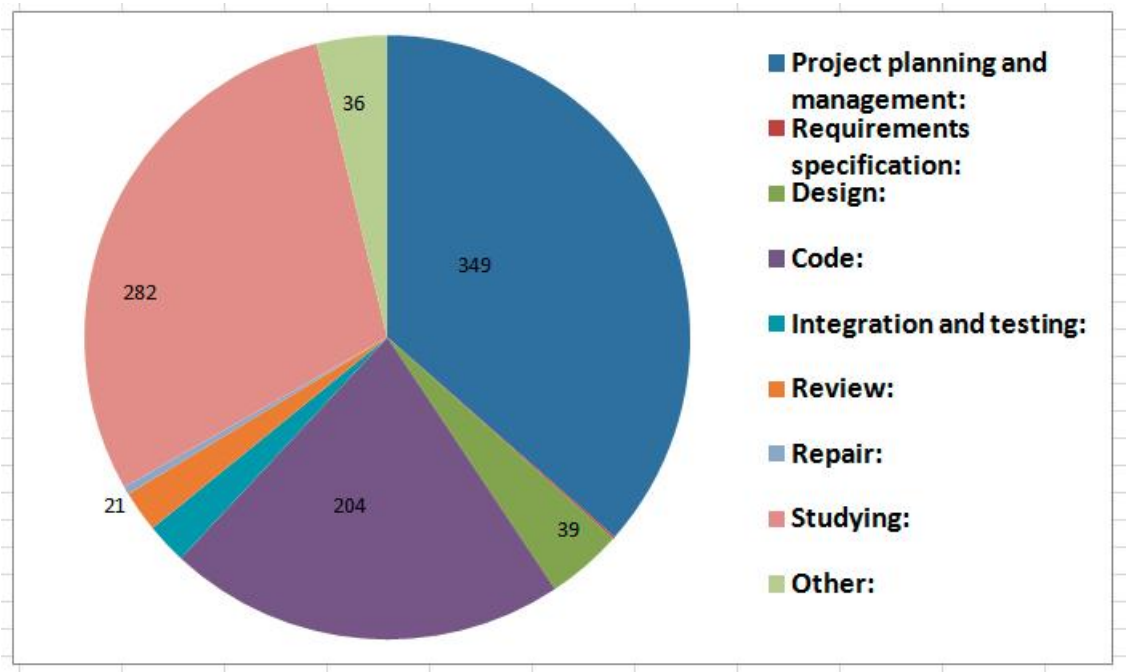


Table 5: Working Hours in a diagram.

Haptic Paintball +

Overview

The Haptic Paintball + project is a continuation of the Innosummer 2010 Haptic Paintball project. The goal of the project was to design and create a software application that uses the positioning and sensor technologies. In the beginning of the project we have analyzed the legacy code and resources left to us by the previous project. Then we had a lot of brainstorming sessions to determine the directions we will be going both in technology and in design.

The end result is Commander, an experimental spatial real-time mobile strategy game incorporating innovative gameplay via haptics and player positioning, developed with Qt. Players compete against each other for world domination by capturing and defending cities around the world.

Games begin with 14 randomly selected cities displayed on a world map. Each player has control of a single city and begins capturing neutral cities to take control of them. Each city under the control of a player generates units at a fixed rate relative to the real-world population of the city. These units can be used to capture other cities or they can be left to defend their current city. A city is captured by being attacked by a number of units greater than the number currently defending it. Once a city is captured, its ownership changes and it begins generating units for its captor.

Players have three commands at their disposal: attack, deploy and nuclear attack. "Attacking" sends units from a controlled city to an enemy city in an attempt to weaken or capture it. Each attacking unit "kills" a defending unit at a 1:1 ratio. For example, if a base holding 42 units is attacked by an enemy with 34 units, 8 units will remain at the base and ownership of the city will not change. The base will continue regenerating units. "Deploying" moves units between cities owned by the same player in order to increase the defense of a city. "Nuclear attacks" work the same way as attacking, but special nuclear missiles are fired at cities that destroy all the units present and return the city to a neutral state. Every minute, a nuclear missile randomly appears on a city. The controller of the city may fire the nuke at will. A city captured with an unfired nuke forfeits the nuke to the enemy.

To perform any of the aforementioned commands, a player moves to a physical location within proximity of the HAIP beacons. Player locations are tracked using HAIP tags they carry on their body. Their movement and location is plotted on the map on their N900 mobile device. When their location is on a city they control on the map, they can attack, deploy or fire a nuke. Each command is activated by performing a different hand gesture while holding an Ariane sensor box in the direction of a city. The location- and rotation-aware sensor boxes know the direction users are facing when issuing commands.

The game ends when one player controls all cities on the map.

Organization and management

Project managers

Olli Alatalo

Mikhail Kapitonov

Project crew

Christopher Contolini

Jenna-Riia Karhunen

Massimo Prencipe

Miika Mikkola

Sunil Chaudhary

Tejal Mate

Tomi Fagerlund

Client

Antti Salomaa (Demola project coordinator), Bernard Garvey (Hermia), Dari Trendafilov (Nokia) and Vilja-Kaisa Aaltonen (Nokia)

Others

Antti Koskenalho (Previous HP-team member) and Jukka Peltomäki (Previous HP-team member)

Project team members: Tomi, Chris, Sunil and project managers Olli, Mike.

Methods and tools

Development tools

The game runs on the Maemo OS in Nokia N900 devices with 800x480 resolutions. For programming in Qt 4.6, the developers used mostly the Qt Creator IDE. For documentation and tracking sprint backlogs we used a wikimedia installation and Google Docs. Redmine was used for issue tracking and Mercurial for revision control. Ariane Sensor Boxes input user hand movements HAIP beacons tracked player locations.

Development model and other methods

The project was developed under modified scrum and sprint like weekly cycle. As the team worked individually most of the times there were no daily stand-offs. Project managers performed the task of scrum master, to remove any obstacles faced by the team member while doing their tasks.

Testing tools and methods

Testing was at the beginning planned to be quite formal Scrum-style testing process, but as the project went on forward it became quite informal and not much of it was reported. At the beginning there were plans to try out an open sourced Qt test automation tool called Testability Driver for functional test automation, but the immature state of the tool made it quite impossible to use.

Project phases and development model

Planning and Design

In the planning stage we had a lot of brainstorming sessions. The planning could be divided into three parts: project planning, technological design and conceptual design. The project planning part was relatively simple: we just took Scrum and modified it a little for our purposes, then wrote down rough Sprint start and end dates and the overall schedule.

Implementation

During the implementation it became apparent that our intensive workshops do not go well with the Sprint planning. We practically had one long sprint in which we tried to implement as much features a possible. Tasks were assigned dynamically and results were checked on the spot.

Testing

The code was tested during the implementation stage while we were doing iterative development of the game. We attempted to play a match almost during every workshop and then fixed whatever errors we encountered. This was also done to evaluate the quality of gameplay. Project itself was tested when it was finished.

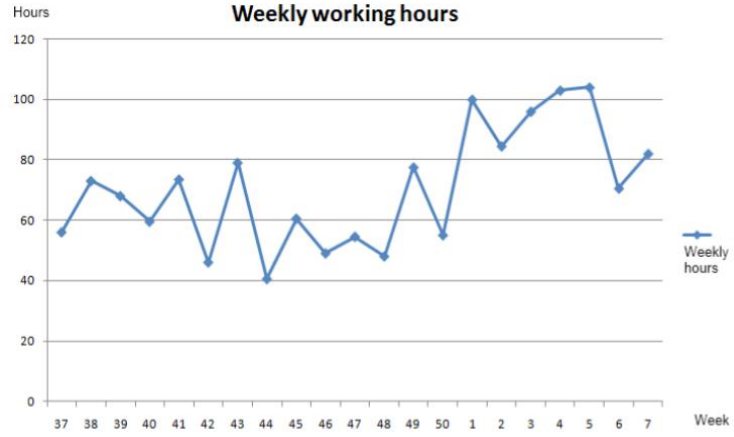
Experiences

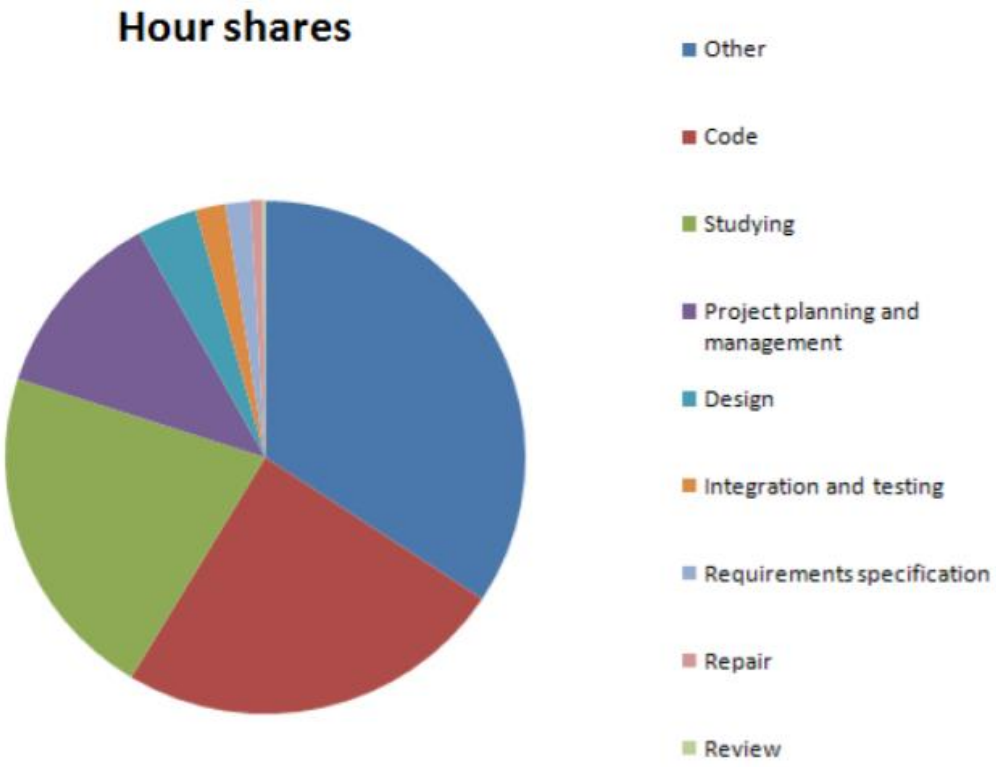
The project was a good exercise in agile development amongst a team of competent peers. We learned how difficult game design can be -- it's a challenge to make software "fun".

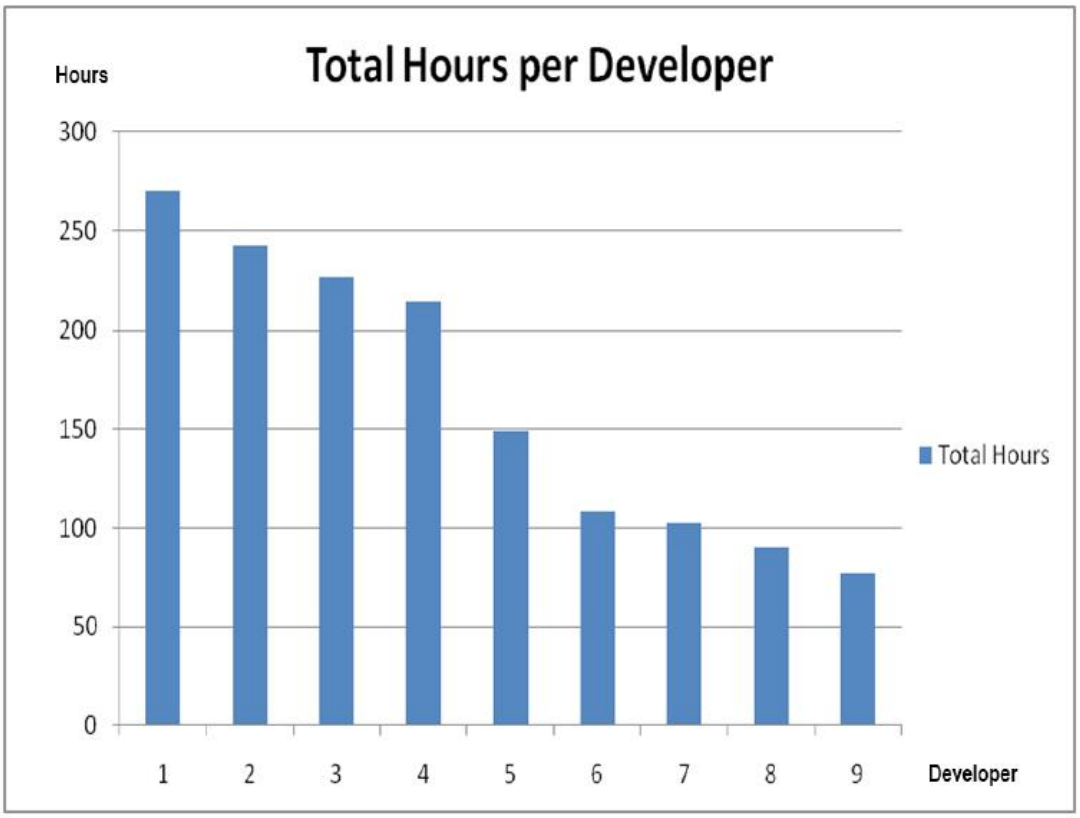
The project was an enjoyable experience because we had fun working with like-minded peers. Difficulties arose with the prototype technologies but we were interested in using bleeding edge technologies. The international composition of the group also made it an interesting experience. Making the gameplay video was a lot of fun.

Working with Qt was a foreign experience for some of us, including the C++ language. Fortunately, our lead developer had significant C++ experience. This was the first time any of us had worked on mobile devices -- learning to design for such a small screen came with its challenges.

Statistics






Total Hours per Developer
Hours
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
9
Developer
Total Hours

Ubiquity

Overview

“Ubiquity” formerly “Ubiquitous Personal Data Storage” has the aim to develop an end-to-end backup system for the user data scattered across various devices like mobile phones, personal computers, network drives and various internet services. This has been built upon an existing system which enables users to share their personal content in a distributed and centralized manner.

Few components ie Visual ReST server, Back-up service, storage devices and user's devices comprise the system. Visual ReST server, developed in Tampere University of Technology, stores the metadata of user content while the actual content could be in user device devices. The back-up service which is the development target of this project, gets meta-data from Visual ReST server and initiates transferring to-be-backed-up content from user devices to user configured remote storage components such as a network file system or cloud storage. Backup service can be controlled by the user to configure backup schedule, to select contents that need backup after authentication and accordingly file transfer from user device to remote storage is initiated. The service interacts with Visual ReST server and storage interface. In brief, the backup service authenticates valid user, stores users data and detects storage devices and initiates the communication between user's own device and user configured data storage devices. Though it was proposed to implement backup system considering firewalls and NATs, the project has to bypass the idea for the sake of simplification of project goal.

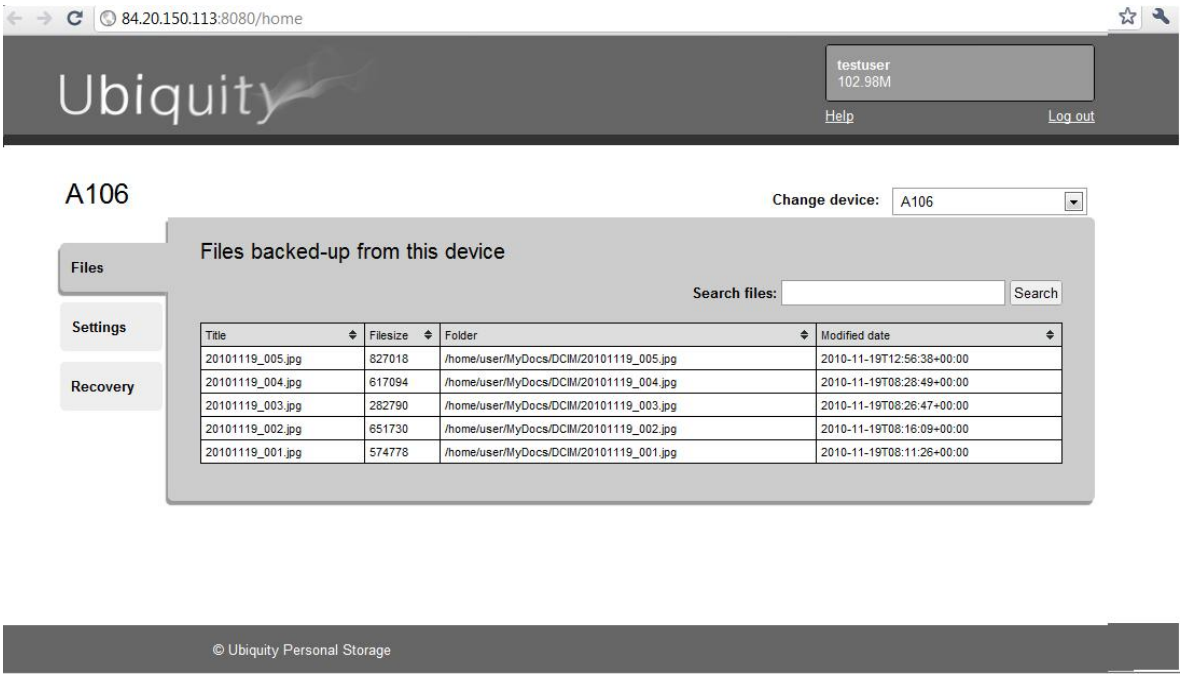


Fig 1: Screen shot of UI part

Organization and Management

Project Members

Avishek Barua	Manager Role
Saravanan Dhanabal	Manager Role
Reza Ahliaraghi	Developer
Fareed Ahmed	Developer
Juha Pentinmaki	Developer
Christopher Contolini	User interface and usability
Helena Hornborg	User interface and usability
Jussi Palomaki	User interface and usability

2.2 Client

Timo Aaltonen, Tapani Leppanen from Nokia Research Centre

2.3 Co-ordination

Bernard Garvey in Demola

2.4 Other stakeholder

Timo Poranen Course Lecturer, Department of Computer Sciences, UTA

Methods and tools

The project was developed with python using the framework Django. Python version used in the development is 3.1. Version control was handled with Tortoise SVN. The server support for version controlling was available in Demola. Google docs was used to collect working hours from team members. Project info and documents were stored in project wiki maintained in Google sites. For API documentation for Python modules Epydoc was used.
Communication and almost all formal information were shared through email. Skype and cell phone were even used to attend weekly meeting from distant places.

Nokia N900 phones were provided from Demola to use in the testing and other phases of the project.

Project phases and development model

The development model used was modified Scrum. Sprints were about 3 weeks long though at first it was planned to be of 2 weeks and only sprint 0 was of two weeks. There were 5 iterations in the project. The project group met weekly and communicated with e-mail or skype.

Here in the chart below spring arrangements are described.

Sprint	Date
Sprint 0	11.10.2010 - 25.10.2010
Sprint 1	25.10.2010 - 15.11.2010
Sprint 2	15.11.2010 - 06.12.2010
Sprint 3	06.12.2010 - 03.01.2011
Sprint 4	03.01.2011 - 24.01.2011
Sprint 5	24.01.2011 - 14.02.2011

Here are most important dates of the project. In Dec, 2010 no review meeting was held as most of team members were on holiday.

Reviews	Date
Project Preliminary Ananlysis	24.09.2010
Project Plan	25.10.2010
Review Meeting	23.11.2010
Review Meeting	29.11.2010
Review Meeting	31.01.2011
Review Meeting	21.02.2011
Review Meeting	28.02.2011
Final Meeting	18.03.2011

Experiences

The reluctance in the developers to take python as development tool was for the risk as it was new to all and they were interested in some other tools. Even the idea was complex to implement. In spite of that it was really a daunting task to implement a complex system with a new learned tool. Considering the difficulties and challenge we thank to our developers to make the project successful.

We had a great experience to work with real client like Nokia. We thank them and also thank to Demola for their assistance in all through the way. It was a project in which we experienced the issues that arise in real project and we decide ourselves, assist each other in a team. We faced difficulties; and we explored the path for resolution. It is a good motivation indeed which will make us able to work in future projects.

Statistics

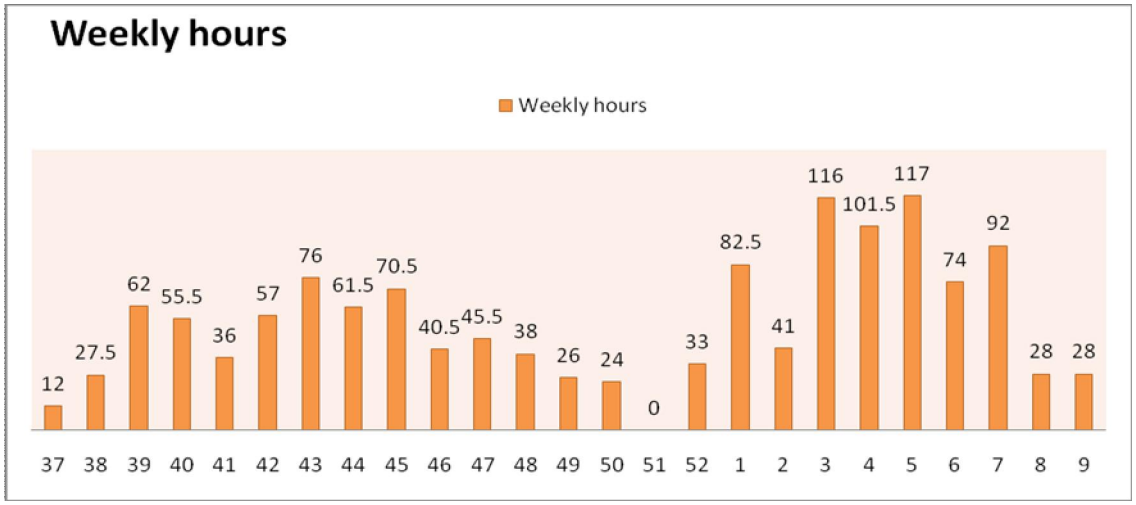


Table 1: General project information.

Team size	Dev. model	Start date	End data	Weeks	Hours
2+3+3	Scrum	17.09.2010	28.02.2011	24	1345

Table 2: Group effort by activity.

Activity	Planning and management	Req. Spec.	Design	Code	Integra. and testing	Reviews	Repair	Study	Other	Total
Hours	295	34	29	288	60	27	41	118	54	946
%	31.18	3.59	3.07	30.44	6.34	2.85	4.33	12.47	5.71	100%
Usability	58.5	3	38	51	0	0	0	16	232.5	399
Total	353.5	37	67	339	60	27	41	134	286.5	1345

Table 3: Requirements and high-level design outcomes.

Number of requirements	**UI screens**
17	3

Table 4: Documents.

Document	**Pages**	**Versions**
Preliminary Analysis	5	1.0
Requirements Specification	4	1.0
Final Report	12	1.0
Usability Interface Plan	14	1.3
Ubiquitous BS Mockup	1	4.0
User Interface Document	17	2.0
Test Plan	7	1.0
Project Story	7	1.0
Weekly reports	19	-

Table 5: Codelines.

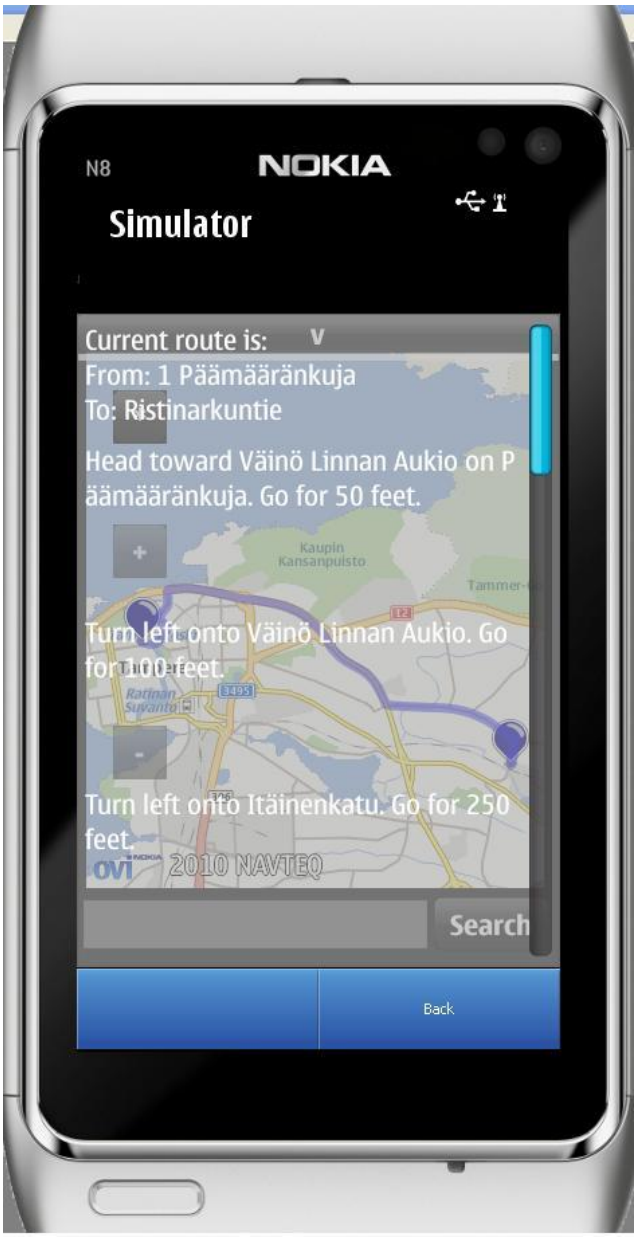
Language	**LOC**	**External Source code**	**SLOC**
Python	931	2859	3790
Javascript	600		1025
HTML			18919
CSS			1352

Mobile Transport

Overview

The product of this project is a demo application, which provides the TKL (Tampereen Kaupunkiliikenne Liikelaitos) bus schedules as well as routes information to the user by getting users location from GPS. The product runs on the mobile device so it is in the touch of the users at all times. Our product tries to make public transportation more user friendly.

User can use application to search routes, and see which buses user needs for traveling to his destination. User can also save his favourite routes and check current timetables for those.



Organization and management

Project Manager:

Antti Leppänen

Team members (developers):

Nikita Ishkov

Timo Virtanen

Tek Prasad Gautam

Usability team member:

Antti Suoninen

From left to right: Antti, Nikita, Timo, Antti S, Tek.

Methods and tools

Mobile phone client was done by using QtSDK and programming language was C++. QtMobility was used to get GPS position of the device. Communication between client and server uses HTTP protocol and data is packaged in JSON format.

Server side is using PHP and Symfony framework.

A wiki was used for communication, keeping track of schedule, sharing of documents and distributing other information. The wiki software in use is MediaWiki and it is installed on the virtual server provided by Demola (http ://84.20.150.111/mediawiki/). We used git for version controlling which is also running on this server. Hour tracking is done with Google Docs by using form that team members can easily fill.

The tools used for documents are Microsoft Word and OpenOffice.org Writer. Spreadsheets are maintained by Microsoft Excel and Google Spreadsheets.

Project phases and development model

We started project by trying to follow guidelines of Scrum. Our plan was to have 2-3 week sprints. We didn't manage to follow Scrum's quite strict guidelines because our project got in very chaotic state because team members started to disappear and taking new members seems like a mistake now. None of the new members which we got didn't finish project. After quite long pregame and planning phase we managed to have only 2 sprints and then we stopped following Scrum. We moved to more agile "just code"-mode without any strict plans for sprints. Our goal was to finish as many tasks we ever could do for the next meeting with our client. This way we started to get results and we begun to move out of the chaos to more stable ground.

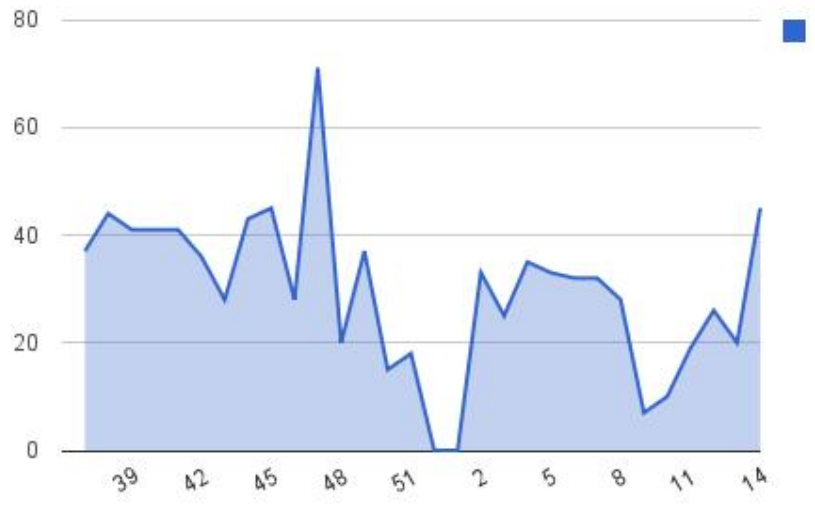
Experiences

- Team members quitting during project. We had serious problems because we lost so many team members during project (1+3). That affected motivation of remaining members quite badly and it took some time to get things normalized. Better project management might have helped to speed this normalization process.
- Technical problems. QtMobility was at quite early stage in the beginning of the project and that slowed down project's start. We also had to wait for new QtSDK which supported N8.
- Uneven hours. Some team members didn't have enough skills to help enough in implementation of the project so the more skilled team members had to work more hours.
- Communication. It's sometimes hard to communicate and make sure that everyone understands things in multicultural team.

What to do better

- Risks management, resources technology, what to do when everything goes wrong.
- More prototyping
- It wouldn't hurt if each team member would know or were willing to learn even the tiniest bit of each technology used. That could minimize the risk of someone not being able to do his share of work for some time, making it still possible to move the project forward - even if slower.

Statistics



Team size	Dev. model	Start date	End data	Days	Hours
1+3+1	SCRUM	6.9.2010	25.4.2011	232	851

Table 1: General project information.

Activity	Planning and management	Req. specification.	De-sign	Code	Integration and testing	Reviews	Repair	Study	Other	Total
Hours	206,5	11	33,5	272,5	51	9	0	144	152	879
%	24	1	4	30	6	1	0	17	17	100 %
Usabi-lity	-	-	-	-	-	-	-	-	-	184
Total										**1063**

Table 2: Group effort by activity.

Document	Pages	Versions
Preliminary analysis	7	2
Project Plan	28	2
Usability analysis	8	3
Requirements specification	-	-
Design plan	-	-
User interface document	8	5
Test plan	-	-
Test report	-	-

Usability test report	9	1
Final report	12	-
Project's story	3	-
Weekly reports	28	-

Table 3: Documents.

Language	C++, PHP
LOC	3155
SLOC	2329
Reused code	0
Reused and modified	0
Classes	-
Functions	-
Code revisions	-

Table 4: Codelines.

TamBiC2

Overview

Over the years, Mr. Cooper has carefully selected and gathered a collection of texts in both English and Finnish language. Accompanied by their corresponding translation texts, these texts form two separate corpora — one with originally English texts with their Finnish translations, and one with originally Finnish texts with their English translations — which are stored in a database. From there they can be access with an online search application — TamBiC, a search engine that was designed by a project group in the Department of Computer Sciences last year (2009–2010). The English philology students in Tampere University use this search engine to gather research material, for example on translation practices concerning certain words or phrases (like how word “play” is translated in Finnish in different contexts), for their Master’s Thesis works.

This year (2010-2011) TamBiC2 project added some new features and make some user interface enhancements.

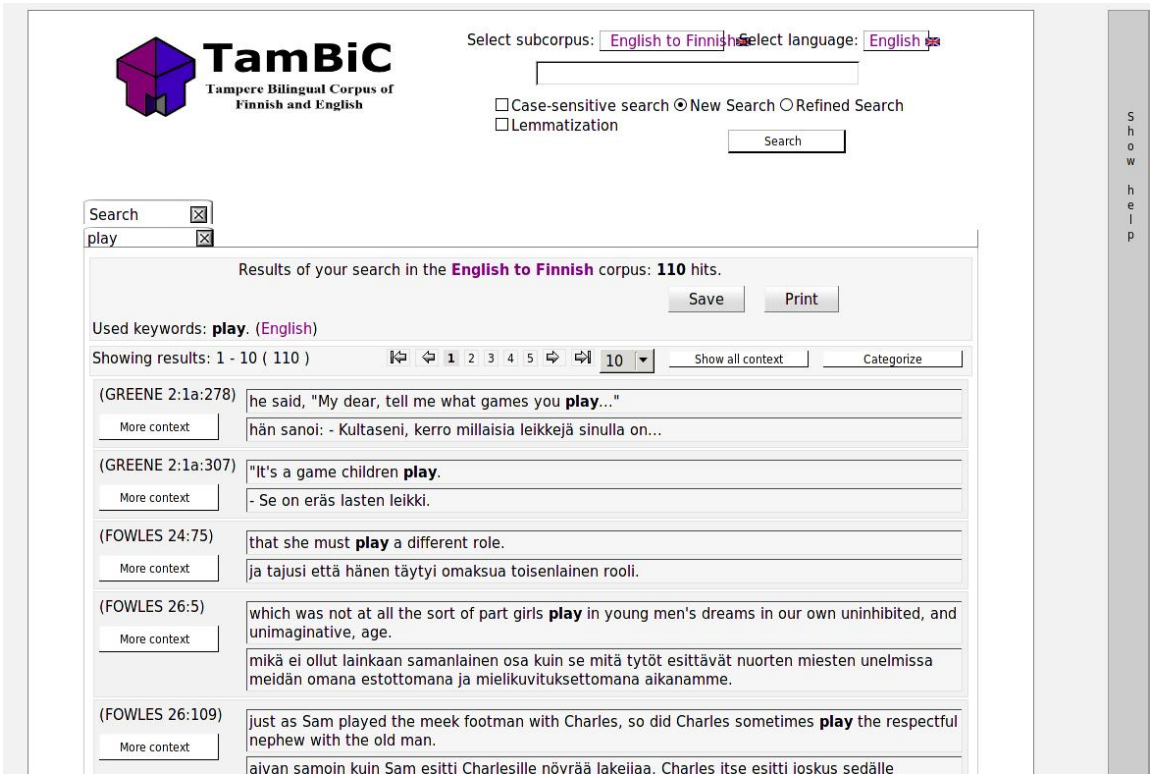


Figure 1: New tambic.

Organization and management

Manteli Numminen, project manager

Tommi Kallio, project manager

Pablo Perez Garcıa, developer

Joanes Errea Iribarren, developer

Hanna Ingalsuo, developer

Jere Myyryläinen, developer

Mikko Pajulahti, usability expert

Francisco Luis Pelluz Jimenez, developer

Every monday were weekly follow up meetings to talk about project related issues and see the progress of the actual implementation. Also extra meetings on Fridays took place, to solve questions and doubts about the assigned work. In every meeting most of the team members where present, as well as the usability expert.

Methods and tools

The project work was guided using project's private forum, a site in address http://www.teamspace4students.com/login.html. The web site contains up-to-date lists on what to do, working hour report, contact information, project schedule, a discussion forum, and all the project documents. Only our team members and course director Timo Poranen have access to the forum.

Our programming environment was Eclipse PHP version (i.e. "Eclipse for PHP Developers"), which also had plug-in to the Subversion tool we use for the version control of the code files. Eclipse platform versions 3.5.0 was used (where the user UI versions is 1.1.0).

The common TamBiC code file repository with browsing capability is located in https://svn.cs.uta.fi/tambic, powered by Subversion (version 1.6.5).

The department of the Computer Science has granted us a working space in one of its Linux servers (cluster.cs.uta.fi), where we can set up our TamBiC corpus database, needed web pages and code files for development purposes.

Programming languages used to create the web application was HTML (for creating the basic web page structure), CSS (for defining the generic outlook of the web pages) and PHP (which provides the needed functionality – e.g. making queries etc.).

The database built to the cluster server was PostgreSQL type, which is a widely-used, open- source object-relational database system, released under BSD-style license.

Project phases and development model

Work plan was to follow the iterative and incremental development model. In iterative model the project work will be carried out in several iterative steps, each of which may lasts weeks or even months.

Each iteration round starts with planning the content of the iteration: a couple of desired features was selected to be implemented, and the needed implementation work and testing are planned. After that, the project team implement the features, and continue to integrate the new software into the running system, and tests the implementation, as well.

No.	Week	Tasks	Other things
	37–38	Starting. Group organization.	
1	39–50	• UI design • Lemmatization • Elimination untranslated sentences in the search results • Bracketing ("(word OR another) AND search") • Update help file	• 17.9 First client meeting • 24.9 Preliminary analysis meeting • 8.10 Project plan meeting • New members Hanna Ingalsuo and Joanes Errea Iribarren
	51–52,1	Christmas holiday	
2	2–9	• UI design • Lemmatization • Search inside multiple search results • Categorization search • Part of speech	• 4.3 Client meeting
3	10	End of project!	X.X Final meeting

At the end of the iteration round, the results of the iterative work are evaluated with the client. We tried to implement about four to five iterations, but it turned out that we did one big iteration. There were many reasons that lead this, but main reason was that there was no pressure to give increment results. TamBiC was implemented last year and it works very well.

Experiences

The risks assessed in the beginning of the project are listed in the project plan. Below are the risks that actualized during the project and how they were handled. Risks are sorted by their nature.

No	Risk	Likelihood	Consequences	Threat	Actions to take
1	Quitting project members	Possible	Substantial	Major	1. Attractive work environment, including challenging and rewarding work. 2. Participation of all members in most of the tasks.
3	Editing the old code causes problems.	Possible	Medium	Medium	Use enough time to understand the old code.

Quitting project member didn't harm that much, because we got two additional developers. Also quitting happened at early stage.

Statistics

Team size	Dev. model	Start date	End data	Days	Hours
2+5+1	Incremental	06.09.10	10.03.11	136	1023

Table 1: General project information.

Week	PP&Man	RS	Des	Code	Int&Test	Review	Repair	Study	Other	**Total**	Accumulated	Goal
36	**5**	**0**	**0**	**0**	**0**	**0**	**0**	**7**	**3**	**15**	15	65.6
37	**15**	**0**	**0**	**0**	**0**	**0**	**0**	**12**	**4**	**31**	46	131.2
38	**24**	**0**	**0**	**0**	**0**	**0**	**0**	**12**	**2**	**38**	84	196.8
39	**22**	**0**	**2**	**0**	**0**	**0**	**0**	**14**	**3**	**41**	125	262.4
40	**21**	**0**	**3**	**0**	**0**	**0**	**0**	**8**	**2**	**34**	159	328
41	**10**	**0**	**0**	**0**	**0**	**0**	**0**	**3**	**1**	**14**	173	393.6
42	**4**	**0**	**6**	**0**	**0**	**0**	**0**	**4**	**3**	**17**	190	459.2
43	**15**	**0**	**1**	**4**	**0**	**0**	**0**	**13**	**1**	**34**	224	524.8
44	**17**	**0**	**5**	**2**	**0**	**0**	**0**	**11**	**1**	**36**	260	590.4
45	**13**	**0**	**8**	**6**	**0**	**2**	**1**	**9**	**2**	**41**	301	656
46	**16**	**0**	**5**	**13**	**0**	**3**	**0**	**13**	**8**	**58**	359	721.6
47	**17**	**0**	**4**	**4**	**0**	**0**	**0**	**6**	**2**	**33**	392	787.2
48	**22**	**0**	**5**	**4**	**0**	**6**	**0**	**12**	**10**	**59**	451	852.8
49	**5**	**0**	**0**	**6**	**0**	**1**	**0**	**6**	**7**	**25**	476	918.4
50	**9**	**0**	**4**	**3**	**0**	**2**	**0**	**9**	**3**	**30**	506	984
51	**7**	**0**	**3**	**1**	**0**	**0**	**0**	**5**	**2**	**18**	524	1049.6
1	**1**	**0**	**5**	**0**	**0**	**1**	**0**	**1**	**1**	**9**	533	1115.2
2	**4**	**0**	**3**	**6**	**0**	**0**	**2**	**6**	**3**	**24**	557	1180.8
3	**17**	**0**	**6**	**2**	**1**	**1**	**0**	**8**	**6**	**41**	598	1246.4
4	**17**	**0**	**4**	**13**	**3**	**6**	**1**	**10**	**4**	**58**	656	1312
5	**16**	**0**	**1**	**15**	**7**	**3**	**5**	**7**	**7**	**61**	717	1377.6
6	**13**	**0**	**11**	**18**	**4**	**1**	**1**	**12**	**7**	**66**	783	1443.2
7	**27**	**1**	**4**	**14**	**5**	**4**	**2**	**6**	**12**	**75**	858	1508.8
8	**24**	**0**	**0**	**15**	**7**	**7**	**3**	**7**	**12**	**77**	935	1574.4
9	**17**	**1**	**1**	**12**	**5**	**5**	**4**	**5**	**5**	**55**	990	1640
Total	**360**	**2**	**81**	**138**	**31**	**42**	**19**	**206**	**111**	**1023**		**617**
											Total hours (estimate)	1640

Table 2: Group effort by activity.

Document	Pages	Versions
Preliminary analysis	4	1
Project Plan	22	3
User interface document	2	1
Final report	15	1
Project's story	5	1
Weekly reports	25	1

Table 3: Documents.